

Department of Informatics  
University of Fribourg (Switzerland)

---

# Neural Machine Reading for Domain-Specific Text Resources

---

THESIS

presented to the Faculty of Science and Medicine of the University of Fribourg (Switzerland)  
in consideration for the award of the academic grade of  
*Doctor of Philosophy in Computer Science*

*by*

Sebastian Arnold

*from*

Berlin, Germany

Thesis N° 2220  
Print Seven  
2020

Accepted by the Faculty of Science and Medicine of the University of Fribourg (Switzerland)  
upon the recommendation of:

Prof. Dr. Philippe Cudré-Mauroux, University of Fribourg (Switzerland), thesis supervisor,  
Prof. Dr.-Ing. habil. Alexander Löser, Beuth University of Applied Sciences Berlin (Germany),  
thesis supervisor,  
Prof. Dr.-Ing. Laura Dietz, University of New Hampshire (USA), external examiner,  
Prof. Dr. Ulrich Ultes-Nietsche, University of Fribourg (Switzerland), president of the jury.

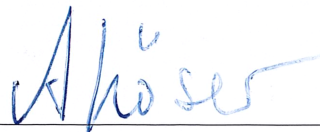
Fribourg, October 12, 2020

Thesis supervisor

A handwritten signature in blue ink, appearing to be 'P. Cudré-Mauroux', written over a horizontal line.

Prof. Dr. Philippe Cudré-Mauroux

Thesis supervisor

A handwritten signature in blue ink, appearing to be 'A. Löser', written over a horizontal line.

Prof. Dr.-Ing. habil. Alexander Löser

Dean

A handwritten signature in blue ink, appearing to be 'G. Rainer', written over a horizontal line.

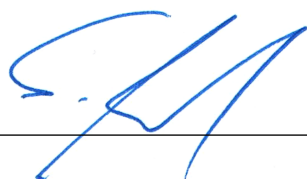
Prof. Dr. Gregor Rainer

# Declaration of Authorship

Title: "Neural Machine Reading for Domain-Specific Text Resources"

I, Sebastian Arnold, declare that I have authored this thesis independently, without illicit help, that I have not used any other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Signed:



Date:

08.09.2020



*Die Natur muß gefühlt werden, wer nur sieht und abstrahirt, kann ein Menschenalter, im Lebensgedränge der glühenden Tropenwelt, Pflanzen und Thiere zergliedern, er wird die Natur zu beschreiben glauben, ihr selbst aber ewig fremd sein.*

— Alexander von Humboldt, an Johann Wolfgang von Goethe, Paris 3. Januar 1810.



# Acknowledgements

The journey I took towards this PhD thesis not only required from myself to grow a rough idea into a larger vision. It also required me to step back and focus on precise scientific contributions over and over. Most of all, however, it required me to ask the people around me for help.

In particular I would like to thank my supervisors Alexander Löser and Philippe Cudré-Mauroux. Alexander guided and facilitated my research since the very start of my Bachelor's thesis at TU Berlin, and will continue to be a signpost for my own visions. Philippe helped me to establish the groundwork for my contributions and responded to all my concerns and questions on the way with comprehensive support from Fribourg.

I am very fortunate to be in company of many supportive mentors from the Data Science Research Center at Beuth University of Applied Sciences Berlin. Felix A. Gers organized our journal club discussions, provided methodical soundness and the right amount of appreciation in any situation; Peter Tröger not only established an excellent technical infrastructure that made it possible to concentrate on my work, but also contributed precise comments on the group's visions; Amy Siu helped me many times with concise feedback for my research proposals.

Most of this work would not have been possible without my coauthors, students, discussion partners and colleagues Torsten Kiliyas, Robert Dziuba, Rudolf Schneider, Christopher Kümmel, Robin Mehltitz, Tom Oberhauser, Betty van Aken, Benjamin Winter, Paul Grundmann and Michalis Papaioannou (in order of appearance), as well as the entire DATEXIS and eXascale Infolab teams. I would further like to thank Djellel E. Difalla for his support in the first phase of this thesis; Iryna Gurevych for introducing me to the Natural Language Processing community; and Laura Dietz for being member of my committee and providing her constructive feedback during the writing of this thesis.

I would like to express my personal gratitude to Franziska, Leonora and my family, Ms. Bräutigam, my friends and fellow musicians, who provided me with emotional grounding and creative space for the ups and downs during these five years. I would like to thank everyone who accompanied me along this way and made this journey possible.





# Abstract

The vision of Machine Reading is to automatically understand written text and transform the contained information into machine-readable representations. This thesis approaches this challenge in particular in the context of commercial organizations. Here, an abundance of domain-specific knowledge is frequently stored in unstructured text resources. Existing methods often fail in this scenario, because they cannot handle heterogeneous document structure, idiosyncratic language, spelling variations and noise. Specialized methods can hardly overcome these issues and often suffer from recall loss. Moreover, they are expensive to develop and often require large amounts of task-specific labeled training examples.

Our goal is to support the human information-seeking process with generalized language understanding methods. These methods need to eliminate expensive adaptation steps and must provide high error tolerance. Our central research question focuses on capturing domain-specific information from multiple levels of abstraction, such as named entities, latent topics, long-range discourse trajectory and document structure. We address this problem in three central information-seeking tasks: Named Entity Linking, Topic Modeling and Answer Passage Retrieval. We propose a collection of Neural Machine Reading models for these tasks. Our models are based on the paradigm of artificial neural networks and utilize deep recurrent architectures and end-to-end sequence learning methods.

We show that automatic language understanding requires a contextualized document representation that embeds the semantics and skeletal structure of a text. We further identify key components that allow for robust word representations and efficient learning from sparse data. We conduct large-scale experiments in English and German language to show that Neural Machine Reading can adapt with high accuracy to various vertical domains, such as geopolitics, automotive, clinical healthcare and biomedicine. This thesis is the first comprehensive research approach to extend distributed language models with complementary structure information from long-range document discourse. It closes the gap between symbolic Information Extraction and Information Retrieval by transforming both problems into continuous vector space representations and solving them jointly using probabilistic methods. Our models can fulfill task-specific information needs on large domain-specific text resources with low latency. This opens up possibilities for interactive applications to further evolve Machine Reading with human feedback.



# Zusammenfassung

Machine Reading ist die Vision, Text automatisiert zu verstehen und in maschinenlesbare Form zu überführen. Die vorliegende Dissertation nimmt sich dieses Problems an und legt dabei besonderes Augenmerk auf die Anwendung in Unternehmen. Hier wird häufig eine große Fülle domänenspezifischen Wissens in Form von unstrukturierten Textdaten vorgehalten. Existierende Methoden der Informationsextraktion weisen in diesem Szenario erhebliche Mängel auf. Häufige Fehlerquellen sind heterogene Dokumente, eigentümliche Sprache, abweichende Schreibweisen und verrauschte Daten. Selbst spezialisierte Methoden können diese Herausforderungen nur mit eingeschränkter Trefferquote bewältigen. Zusätzlich sind sie kostspielig in der Entwicklung und benötigen oft große Mengen an annotierten Trainingsdaten.

Unser Ziel ist es, den Anwender im Prozess der Informationssuche mit maschinellen Sprachverständismethoden zu unterstützen. Diese Methoden sollen kostenintensive Anpassungsschritte vermeiden und müssen eine hohe Fehlertoleranz aufweisen. Unsere zentrale Forschungsfrage richtet sich darauf, domänenspezifische Information auf mehreren Abstraktionsebenen zu erfassen. Dies umfasst u.a. die Identifikation von Objekten, latenten Themenverteilungen, Diskursverläufen und Dokumentenstruktur. Im Fokus stehen dabei drei zentrale Prozessschritte der Informationssuche: Named Entity Linking, Topic Modeling und Answer Passage Retrieval. Die vorliegende Arbeit stellt für diese Zwecke eine Sammlung neuronaler Machine Reading Modelle vor. Auf der Grundlage von künstlichen neuronalen Netzen werden hierfür insbesondere Verfahren des tiefen und sequenzbasierten Lernens genutzt.

Das zentrale Ergebnis dieser Arbeit ist eine kontextualisierte Dokumentenrepräsentation für automatisiertes Sprachverständnis, welche in verdichteter Form die Semantik und Grundstruktur eines Textes umfasst. Darüber hinaus werden grundlegende Komponenten vorgestellt, die robuste Wortrepräsentation und effizientes Lernen aus spärlichen Daten ermöglichen. Umfassende Experimente in englischer und deutscher Sprache belegen, dass neuronales Machine Reading mit hoher Präzision auf eine Vielzahl vertikaler Domänen anwendbar ist, wie z.B. Geopolitik, Autoindustrie, Gesundheitswesen und Biomedizin. Diese Dissertation ist der erste umfassende Forschungsansatz, neuronale Sprachmodelle mit komplementären Strukturelementen auf Dokumentenebene anzureichern. Dieser Ansatz schließt die Lücke zwischen symbolischer Informationsextraktion und Informationssuche, indem beide Probleme in kontinuierliche Vektorraumrepräsentationen übersetzt und durchgängig probabilistisch gelöst werden. So können unternehmensspezifische Informationsbedürfnisse mit schnellen Antwortzeiten erfüllt werden. Dies ermöglicht interaktive Anwendungen, die Machine Reading zukünftig mit Hilfe von menschlichem Feedback verbessern können.



# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>Zusammenfassung</b>	<b>xi</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Abbreviations</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The Vision of Machine Reading . . . . .	1
1.2 Domain-Specific Language Understanding . . . . .	4
1.3 Research Objectives . . . . .	6
1.4 Contributions . . . . .	8
1.5 Thesis Outline . . . . .	10
1.6 Publications . . . . .	11
<b>2 Background</b>	<b>13</b>
2.1 Information Extraction . . . . .	13
2.1.1 Text Preprocessing . . . . .	14
2.1.2 Shallow Syntactic Parsing . . . . .	15
2.1.3 Deep Linguistic Processing . . . . .	16
2.1.4 Named Entity Recognition . . . . .	17
2.1.5 Named Entity Linking . . . . .	20
2.2 Distributed Language Representations . . . . .	23
2.2.1 Distributional Hypothesis . . . . .	23
2.2.2 Semantic Vector Space Model . . . . .	24
2.2.3 Neural Distributed Language Representations . . . . .	26
2.2.4 Contextualized Language Models . . . . .	30
2.3 Supervised Sequence Learning . . . . .	33
2.3.1 Finite State Models . . . . .	33
2.3.2 Recurrent Neural Networks . . . . .	35
2.3.3 Long Short-Term Memory . . . . .	35
2.3.4 Transformer . . . . .	37
2.4 Discussion . . . . .	39

<b>3</b>	<b>A Robust Model for Efficient Entity Linking</b>	<b>41</b>
3.1	Introduction . . . . .	42
3.1.1	Task Definition . . . . .	42
3.1.2	Challenges for Entity Linking . . . . .	43
3.1.3	Common Error Analysis . . . . .	43
3.2	Entity Linking Model . . . . .	44
3.2.1	Robust Word Encoding . . . . .	44
3.2.2	Named Entity Recognition . . . . .	45
3.2.3	Named Entity Disambiguation . . . . .	48
3.3	Evaluation . . . . .	49
3.3.1	Evaluation Set-up . . . . .	49
3.3.2	Experimental Results . . . . .	51
3.4	Discussion and Error Analysis . . . . .	53
3.5	Conclusions . . . . .	54
<b>4</b>	<b>Coherent Topic Segmentation and Classification</b>	<b>55</b>
4.1	Introduction . . . . .	56
4.1.1	Challenges for Topic Representation . . . . .	56
4.1.2	Task Definition . . . . .	58
4.2	WikiSection Dataset . . . . .	58
4.2.1	Preprocessing . . . . .	59
4.2.2	Synset Clustering . . . . .	59
4.3	Model Architecture . . . . .	61
4.3.1	Sentence Representation . . . . .	61
4.3.2	Topic Classification . . . . .	63
4.3.3	Topic Segmentation . . . . .	65
4.4	Evaluation . . . . .	66
4.4.1	Evaluation Set-up . . . . .	67
4.4.2	Experimental Results . . . . .	69
4.5	Discussion and Insights . . . . .	70
4.6	Related Work . . . . .	71
4.7	Conclusions . . . . .	74
<b>5</b>	<b>Contextualized Document Representations for Answer Retrieval</b>	<b>75</b>
5.1	Introduction . . . . .	76
5.1.1	Challenges for Answer Passage Retrieval . . . . .	77
5.1.2	Contextual Discourse Vectors . . . . .	77
5.2	Discourse Model . . . . .	78
5.2.1	Entity Representation . . . . .	79
5.2.2	Aspect Representation . . . . .	80
5.2.3	Query Representation . . . . .	81

5.3	Contextualized Document Representation . . . . .	81
5.3.1	Sentence Encoder . . . . .	82
5.3.2	Document Encoder . . . . .	84
5.3.3	Passage Scoring . . . . .	84
5.3.4	Self-supervised Training . . . . .	86
5.4	Evaluation . . . . .	87
5.4.1	Evaluation Set-up . . . . .	87
5.4.2	Experimental Results . . . . .	90
5.5	Discussion and Error Analysis . . . . .	91
5.6	Related Work . . . . .	94
5.7	Conclusions . . . . .	96
<b>6</b>	<b>Systems</b>	<b>97</b>
6.1	TASTY: Interactive Editor for Entity Linking As-You-Type . . . . .	97
6.1.1	Design Challenges . . . . .	97
6.1.2	Interactive Entity Linking Process . . . . .	99
6.1.3	Application Scenarios . . . . .	101
6.2	TraiNER: Bootstrapping Named Entity Recognition . . . . .	102
6.2.1	Active Learning Process . . . . .	102
6.2.2	Graphical User Interface . . . . .	104
6.2.3	Experimental Results . . . . .	105
6.3	Smart-MD: Clinical Decision Support System . . . . .	106
6.3.1	Demonstration Scenario . . . . .	106
6.3.2	Passage Retrieval Process . . . . .	106
6.4	CDV Healthcare Answer Retrieval . . . . .	109
6.4.1	Demonstration Scenario . . . . .	109
6.4.2	Discussion of Healthcare Queries . . . . .	111
6.5	Conclusions . . . . .	112
<b>7</b>	<b>Conclusion and Future Work</b>	<b>113</b>
7.1	Contributions of Neural Machine Reading . . . . .	113
7.2	Review of Research Questions . . . . .	115
7.3	Limitations . . . . .	117
7.4	Business Perspectives . . . . .	118
7.5	Future Work . . . . .	122
	<b>Bibliography</b>	<b>125</b>
	<b>Curriculum Vitae</b>	<b>147</b>





# List of Figures

1.1	Human information-seeking process . . . . .	3
2.1	Part-of-speech and dependency parse tree . . . . .	15
2.2	Mention-entity graph example . . . . .	20
2.3	Syntagmatic and paradigmatic relations . . . . .	24
2.4	Syntagmatic and paradigmatic neighborhood examples . . . . .	25
2.5	CBOW and Skip-gram architectures . . . . .	28
2.6	Paragraph Vector architecture . . . . .	29
2.7	ELMo and BERT architectures . . . . .	31
2.8	Pre-training and fine-tuning procedures for BERT . . . . .	32
2.9	Dependency graph of HMM, MEMM and CRF . . . . .	34
2.10	RNN architecture . . . . .	35
2.11	LSTM memory cell . . . . .	36
2.12	BLSTM architecture . . . . .	37
2.13	Transformer architecture . . . . .	38
2.14	Scaled dot-product attention and multi-head attention . . . . .	38
3.1	TASTY robust letter-trigram word encoding architecture . . . . .	45
3.2	TASTY BLSTM network architecture for Named Entity Recognition . . . . .	46
3.3	Effect of training data size on NER model performance . . . . .	52
4.1	WIKISECTION task overview . . . . .	57
4.2	Training and inference phases of segmentation and topic classification . . . . .	62
4.3	SECTOR neural network architecture . . . . .	63
4.4	Deviation graph of SECTOR topic embeddings . . . . .	66
4.5	SECTOR topic prediction histogram . . . . .	71
4.6	SECTOR topic embedding vector space . . . . .	72
5.1	Structured entity/aspect query example . . . . .	76
5.2	CDV neural network architecture . . . . .	83
5.3	Stages of the passage retrieval process . . . . .	85
5.4	CDV model prediction histogram . . . . .	86
5.5	Comparison of entity/aspect matching errors . . . . .	92
5.6	Comparison of performance across data sources . . . . .	92

5.7	Comparison of aspect prediction performance . . . . .	92
5.8	Comparison of entity and aspect mismatch error classes . . . . .	93
6.1	TASTY graphical user interface . . . . .	98
6.2	Interactive Entity Linking process . . . . .	99
6.3	TraiNER active learning process . . . . .	103
6.4	TraiNER graphical user interface . . . . .	104
6.5	Comparison of sampling strategies . . . . .	105
6.6	SMART-MD graphical user interface . . . . .	107
6.7	Neural network architectures for topic classification and entity recognition . . . .	108
6.8	Visualization of neural topic classification . . . . .	108
6.9	CDV graphical search interface . . . . .	110

# List of Tables

3.1	Experimental results for English news Named Entity Recognition . . . . .	50
3.2	Experimental results for domain-specific Named Entity Recognition . . . . .	50
3.3	Ablation study for Named Entity Recognition . . . . .	51
3.4	Experimental results for English Entity Disambiguation . . . . .	53
4.1	Characteristics of WIKISECTION datasets . . . . .	58
4.2	Distribution of disease headings in the English Wikipedia . . . . .	59
4.3	List of topics in the WIKISECTION dataset . . . . .	60
4.4	Experimental results for topic segmentation and single-label classification . . . .	68
4.5	Experimental results for topic segmentation and multi-label classification . . . .	68
4.6	Experimental results for cross-dataset topic segmentation . . . . .	70
5.1	Distribution of disease headings in the CDV training set . . . . .	81
5.2	Characteristics of CDV training and evaluation data sets . . . . .	88
5.3	Experimental results for healthcare Answer Passage Retrieval . . . . .	90
6.1	Example scenarios for TASTY’s application . . . . .	102
6.2	Examples for CDV healthcare queries and answers . . . . .	110



# List of Abbreviations

## General Abbreviations

<b>e.g.</b>	exemplum gratia (English: for example)
<b>Eq.</b>	Equation
<b>et al.</b>	et alia (English: and others)
<b>i.e.</b>	id est (English: that is)
<b>K</b>	kilo (English: thousand)
<b>M</b>	million
<b>RQ</b>	Research Question

## General Terms and Concepts

<b>AI</b>	Artificial Intelligence
<b>CDSS</b>	Clinical Decision Support System
<b>CPU</b>	Central Processing Unit
<b>CUI</b>	Concept Unique Identifier (in UMLS)
<b>DDx</b>	Differential Diagnosis
<b>DE</b>	German (Deutsch)
<b>EBM</b>	Evidence Based Medicine
<b>EDRM</b>	Electronic Discovery Reference Model
<b>EHR</b>	Electronic Health Record
<b>EN</b>	English
<b>ESI</b>	Electronically Stored Information
<b>ETL</b>	Extract–Transform–Load
<b>GPU</b>	Graphics Processing Unit
<b>H&amp;P</b>	History and Physical Examination
<b>HCI</b>	Human–Computer Interaction
<b>ICD-10</b>	International Classification of Diseases (Version 10)
<b>ID</b>	Identifier
<b>ML</b>	Machine Learning
<b>NIH</b>	National Institutes of Health
<b>NLP</b>	Natural Language Processing
<b>OCR</b>	Optical Character Recognition

<b>PICO</b>	Patient–Intervention–Comparison–Outcome
<b>SCRM</b>	Supply Chain Risk Management
<b>UID</b>	Unique Identifier
<b>UMLS</b>	Unified Medical Language System

### **Natural Language Processing**

<b>BIOES</b>	Begin–Inside–Outside–End–Single
<b>BOW</b>	Bag-Of-Words
<b>CBOW</b>	Continuous Bag-Of-Words
<b>IDF</b>	Inverse Document Frequency
<b>IE</b>	Information Extraction
<b>IR</b>	Information Retrieval
<b>KB</b>	Knowledge Base
<b>KBP</b>	Knowledge Base Population
<b>LM</b>	Language Model
<b>MR</b>	Machine Reading
<b>MRC</b>	Machine Reading Comprehension
<b>NED</b>	Named Entity Disambiguation
<b>NEL</b>	Named Entity Linking
<b>NER</b>	Named Entity Recognition
<b>NIL</b>	Non-Linkable Entity
<b>NL</b>	New Line
<b>NLU</b>	Natural Language Understanding
<b>OOV</b>	Out-Of-Vocabulary
<b>POS</b>	Part-Of-Speech
<b>QA</b>	Question Answering
<b>RE</b>	Relationship Extraction
<b>SRL</b>	Semantic Role Labeling
<b>TDT</b>	Topic Detection and Tracking
<b>TF</b>	Term Frequency
<b>WSD</b>	Word Sense Disambiguation

### **Machine Learning Architectures and Algorithms**

<b>ADAM</b>	Adaptive Moment Estimation
<b>BPTT</b>	Backpropagation Through Time
<b>BLSTM</b>	Bidirectional Long Short-Term Memory
<b>CNN</b>	Convolutional Neural Network
<b>CRF</b>	Conditional Random Fields

<b>DNN</b>	Deep Neural Network
<b>FF</b>	Feed-Forward Neural Network
<b>HMM</b>	Hidden Markov Model
<b>LDA</b>	Latent Dirichlet Allocation
<b>LSA</b>	Latent Semantic Analysis
<b>LSTM</b>	Long Short-Term Memory
<b>MAP</b>	Mean Average Precision
<b>MEMM</b>	Maximum-Entropy Markov Model
<b>PCA</b>	Principal Component Analysis
<b>P@k</b>	Precision at Top- $k$
<b>PMI</b>	Pointwise Mutual Information
<b>Prec</b>	Precision
<b>Rec</b>	Recall
<b>ReLU</b>	Rectified Linear Unit
<b>R@k</b>	Recall at Top- $k$
<b>RNN</b>	Recurrent Neural Network
<b>SGD</b>	Stochastic Gradient Descent
<b>SVD</b>	Singular Value Decomposition
<b>SVM</b>	Support Vector Machine
<b>TBPTT</b>	Truncated Backpropagation Through Time
<b>TSNE</b>	T-distributed Stochastic Neighbor Embedding
<b>VSM</b>	Vector Space Model

### Model and Framework Names

<b>BERT</b>	Bidirectional Encoder Representations from Transformers
<b>CDV</b>	Contextual Discourse Vectors
<b>DL4J</b>	Deep Learning for Java (Deeplearning4j)
<b>ELMo</b>	Embeddings from Language Models
<b>GloVe</b>	Global Vectors
<b>HAL</b>	Hyperspace Analogue to Language
<b>ND4J</b>	N-Dimensional Arrays for Java
<b>NLTK</b>	Natural Language Toolkit
<b>PV, ParVec</b>	Paragraph Vectors
<b>TASTY</b>	Tag As-You-Type
<b>W2V</b>	Word2Vec Embeddings





## Chapter 1

# Introduction

This thesis discusses the challenges and methods for Machine Reading. The vision of Machine Reading is to automatically understand unstructured text and represent the contained information in a machine-readable format. This knowledge is then presented to a user to fulfill one or more information-seeking tasks. Automatic language understanding is a challenge in particular for the application in commercial organizations. Here, information resources consist of heterogeneous document types with domain-specific language. In this scenario, off-the-shelf models will likely fail to produce appropriate results. This arises the need for specialized methods, which are expensive to develop and often require large amounts of annotated training data. Therefore this thesis focuses on the investigation of robust and general methods which can be built efficiently from available data. Our goal is to develop and evaluate these methods over a broad range of document types, vertical domains and information-seeking tasks.

### 1.1 The Vision of Machine Reading

In corporate information systems, the majority of resources consist of unstructured data, including images, audio, video and text documents [Inmon et al., 2019]. The data produced by consumers and industry is continuously growing, but most of it is not transformed into structured records, which would provide the highest business value. As of 2012, only 3.5% of unstructured data worldwide is analyzed or tagged with meta-data, while 23% of the data could potentially contain valuable information [Gantz and Reinsel, 2012]. This data lake is rapidly growing, and a recent IDC study predicts that by 2025, 80% of worldwide data will be unstructured [Reinsel et al., 2018]. Two of the fastest-growing industries are healthcare with projected 36% growth in data production between 2018–2025 and manufacturing with 30% respectively. Interestingly, in a recent MAPI survey 58% of organizations reported a lack of data resources as the most significant barrier to deploy AI solutions [Atkinson and Ezell, 2019]. Moreover, 52% of organizations reported their uncertainty in implementing specific data-based tasks. Hence, there is a strong need by organizations to transform the information contained in unstructured text resources into universal machine-readable representations of knowledge.

**The need for unsupervised language understanding.** Traditionally, the goal of *Information Extraction* (IE) is to automatically transform unstructured sources into structured or semi-structured data formats and process them in a business logic pipeline [Sarawagi, 2008]. A classic architecture for this process is called extract–transform–load (ETL) and is often used to populate the entity-relationship model in data warehousing [Vassiliadis, 2009]. The most common IE techniques for this pipeline originate from research on Natural Language Processing (NLP). These methods focus on the recognition of topics, named entities, e.g. persons or organizations, and their relationships [Sarawagi, 2008]. This is achieved by understanding the syntax of a document as discrete low-level features, such as part-of-speech tags, dependency parse trees or semantic role labels. One inherent problem of this symbolic approach is that the extracted information is represented as isolated ‘nuggets’ [Etzioni et al., 2006]. These need to be discretely processed for downstream tasks such as Named Entity Linking (NEL), Question Answering (QA) or Information Retrieval (IR). Furthermore, the majority of these models are restricted to a specific task on a specific type of language. They are often based on hardcoded rule sets or automatically learned parameters from hand-labeled training examples [Etzioni et al., 2006]. Thus, expensive human labor is required to transfer them to different languages or more specific domains.

**Machine Reading** (MR) is the vision to overcome these discrepancies by the “automatic, unsupervised understanding of text” [Etzioni et al., 2006]. Independently of the methodical approach, the envisioned properties of a MR system comprise the coherent formation of beliefs, inclusion of background knowledge, the ability for implicit inference and scalability towards an arbitrary number of relations [Etzioni et al., 2006]. Ideally, this should be achieved without the need for hand-labeled training examples. Up to today, such a generalized form of language understanding has not yet been solved entirely. However, important ingredients have appeared in the mid 2010s, when deep neural networks (DNNs) have re-emerged as powerful ML models for image processing [Krizhevsky et al., 2012] and core NLP techniques [Goldberg, 2016]. DNNs encode information as dense representations spanning multiple stacked layers of continuous weights. Layers are connected using nonlinear activations. Therefore, these models are able to express complex functions with a large number of dependencies and produce probabilistic outputs rather than discrete decisions. Stochastic gradient descent (SGD) allows DNNs to be optimized with supervised end-to-end training data, eliminating the need for discrete intermediate representations of linguistic syntax.

The DNN paradigm has changed the construction process of IE models drastically—from a symbolic language-oriented perspective into an empirical data-driven discipline. DNNs are highly dependent on large training data sets. But pre-trained representation layers for background knowledge, such as word embeddings or language models, can drastically reduce the iteration times required to build tailor-made models. A large fraction of these representations can be trained using *self-supervision*. This process generates labels from the text itself without human supervision. Eventually, Machine Reading constitutes the transformation of information-seeking systems from symbolic IE towards distributed semantic representations.

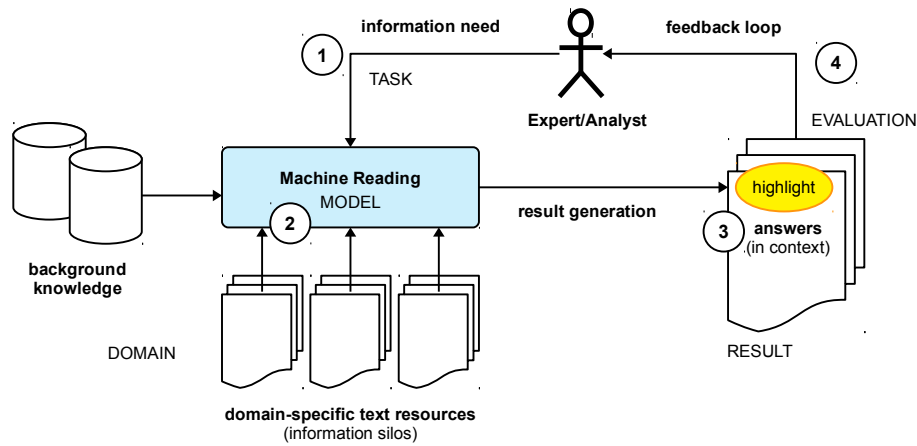


FIGURE 1.1: The process of supporting human information seeking with Machine Reading from domain-specific text resources.

**Supporting human information seeking with Neural Machine Reading.** The information search process is a cyclic activity by a user aiming “to extend his or her state of knowledge on a particular problem” [Kuhlthau, 1991]. Figure 1.1 shows this simplified process in an organization. (1) An expert with an analytical task searches for answers over a large collection of unstructured domain-specific text resources. (2) The MR system formulates its beliefs based on the task, background knowledge and the information contained in the texts. (3) The system presents a coherent result to the user and reasons about its decisions by highlighting the original source context. (4) Finally, the user enters a feedback loop by exploring results, iteratively refining the query, and eventually evaluating the success of the task.

There have been several approaches to describe the complexity of this process. Marchionini [2006] classifies information-seeking activities into three categories: *Lookup* tasks aim to retrieve discrete well-structured objects with high precision (e.g. known item search, navigation, factoid QA). *Learn* tasks require cognitive processing to return aggregated results (e.g. comprehension, comparison, interpretation). Finally, *investigative* tasks are the most complex and require substantial knowledge and integration abilities. These mainly focus on high recall for open-ended search intentions (e.g. discovery, analysis, synthesis, evaluation). With respect to domain-specific applications in the industry, Castro and New [2016] discuss the fundamental functionalities of an artificial intelligence (AI) system as *learning*, *understanding*, *reasoning* and *interaction*. In this thesis, we focus on three central tasks which cover all complexity levels:

- **Named Entity Linking (NEL)** aims to recognize mentions of named entities, such as product, organization or disease names and link them to a structured knowledge base [Bunescu and Pasca, 2006; Hachey et al., 2013; Shen et al., 2015]. Some challenging entity-centric search intentions arise from the scenario. For example: “Which recent news articles mention my product X?”, “Is any of our suppliers mentioned in current risk reports?” or “Show me historical patient records of a cohort suffering from disease Y”. While NEL is a ‘classic’ IE task in the first place, there is a large gap between its current definition as word-level classification task and its application in a real-world information-seeking activity.

For example, mentions might be implicit (e.g. “the former CEO of Apple”), expressed as coreferences (e.g. “our flagship product in 2020”) or can be discussed over a long-range context [Ratinov and Roth, 2009]. Furthermore, many entity types are often overlapping or hierarchically structured [Ling et al., 2015] and it is not clear if a query needs to focus on precise subtypes or more generic supertypes. We will discuss NEL with its challenges and opportunities for a Neural Machine Reading system in Chapter 3.

- **Topic Modeling** aims to discover main themes in a document in order to help organizing a collection [Blei, 2012]. In contrast to named entities, topics are often not explicitly mentioned in the text. Instead, topics are distributions of typical words used to describe a common aspect, e.g. “product features” or “genetics”. Some examples for topical information-seeking tasks are: “Which recent news articles contain technical product reviews?”, “Which risk reports focus on transportation issues?”, “Show me clinical studies that focus on research related to kidneys.” However, Topic Modeling often focuses on capturing entire documents and neglects an opportunity arising from deeper language understanding: Topics might emerge and disappear over the course of long documents, such as research papers that focus on individual aspects in different passages. We will discuss this idea of topic segmentation and classification in Chapter 4.
- **Answer Passage Retrieval** is the task of identifying an answer passage for a given query from a large number of long documents [O’Connor, 1980; Salton et al., 1993]. In contrast to ‘classic’ QA, this task does not focus on factoid questions with short answers. Instead, questions are typically open-ended and answers consist of one or more passages spanning multiple sentences. Example questions are: “Are there any product recalls from company X?”, “Is supplier Y affected by the transportation issues mentioned in the news?”, “What treatment options are currently being researched for IgA nephropathy?”. Answer Passage Retrieval requires the ability to identify named entities and topics in queries and answers, and extract and aggregate passages from long documents that contain coherent answers. We will approach this task in Chapter 5.

There are other potential variations of these tasks, such as Relation Extraction [Sarawagi, 2008], Question Answering [Rajpurkar et al., 2016] or Machine Reading Comprehension [Hermann et al., 2015]. We do not specifically address these tasks in this thesis. In general, all discussed information-seeking tasks can utilize Neural MR as a central component, as depicted in step 2 of Figure 1.1. We will focus on this part of the process throughout this thesis. In the next section, we discuss the requirements and properties of such a MR model.

## 1.2 Domain-Specific Language Understanding

In the information-seeking process, complex problems have to be solved in each of the steps. Especially when applying the tasks to domain-specific text, such as Web blogs, financial risk reports or clinical notes, symbolic IE models will likely fail to produce satisfactory results.

The main reason for this is that language is often ambiguous, and supervised models highly depend on the data they were trained on. Based on prior work on symbolic IE [Löser et al., 2012; Arnold et al., 2014; Maqsdud et al., 2014; Arnold et al., 2015], we identify six central challenges, which we describe below. We will revisit these challenges in Section 7.1.

- **Domain-specific language.** Vocabulary and sentence structure strongly varies between different domains. This is not only the case across different languages (e.g. English or German), but also between different application domains, e.g. geopolitics, manufacturing or healthcare. For example, Wikipedia articles mostly contain generally understandable language. In contrast, financial risk reports or clinical notes might contain a large number of domain-specific abbreviations, idiosyncratic terms and even sentences that do not follow the syntactical rules of language. We require a MR model to be adaptive to domain-specific language. A model trained for one task must be transferable to perform the same task on a different domain without having to rebuild it from scratch.
- **Variations and noise.** Texts may contain spelling errors, morphological variations, incomplete sentences or noise introduced from technical processing, e.g. optical character recognition (OCR) or Web scraping. We require a MR model to be robust towards these variations. The model should prioritize recall over precision and local errors should not affect the model’s performance in the surrounding context.
- **Heterogeneous document structure.** Text resources have a high variance in terms of structure. Documents might be very long (e.g. research articles), medium (e.g. news articles) or short (tweets), strongly structured (e.g. case studies), diverse (e.g. blog articles) or flat (e.g. transcriptions). It is not always possible to identify coherent structural blocks or topics without understanding the text itself. We require a MR model to perform well on this variety of documents without prior training for a specific type. In principle, the model should be able to grasp the overall idea of a text but be sensitive for local information at the same time.
- **High task variance.** Information-seeking tasks span a broad range in specificity and complexity. For example, risk analysts expect high recall for detecting a broad range of possible events. Medical specialists may pose very precise queries that align with a pre-defined taxonomy and focus on rare diseases. Because in many cases we do not know the specificity of a task a priori, it is not feasible to train personalized MR models for specific intentions only. Instead, a well-generalized MR model must support multiple tasks at the same time with zero-shot or only few adaptation steps.
- **Insufficient training data.** In many cases, there exist unlabeled corpora of domain-specific text (e.g. in corporate ‘data silos’), but we do not have access to large amounts of task-specific labeled training data. Labeling data requires expensive human labor and it is not feasible to create large training sets for each specific combination of tasks and domains. Therefore, we aim to focus on efficient models that can be trained with small

amounts of task-specific training data and leverage background knowledge, unsupervised or self-supervised training procedures wherever possible.

- **Error propagation.** Traditional information processing pipelines are sensitive to recall loss, because discrete decisions in early stages of a pipeline (e.g. part-of-speech tagging) propagate into downstream tasks. Typical errors in these stages arise from the ambiguity of language or irregularities in the text. Often, errors are not recoverable in a later stage. Therefore, a MR system should maintain a consistent differentiable model from end to end. This allows probabilistic beliefs to be pushed up to the final decisions at task level. Furthermore, end-to-end approaches may provide important features for error analysis and enable system-wide correction of errors using backpropagation.

**Approach and scope of this thesis.** We approach these challenges by proposing a collection of *Neural Machine Reading* models to address the three central tasks discussed above: Named Entity Linking, Topic Modeling and Answer Passage Retrieval. Our work is built upon the paradigm of deep neural networks. This enables us to design probabilistic models invariant to all languages expressed in Western Latin characters. Furthermore, DNNs allow us to reuse pre-trained components in different architectures. We focus on supervised and self-supervised techniques to train these architectures with end-to-end textual examples, independent of their language or domain. In addition, we address the challenges of robustness and training efficiency throughout this thesis. Consequently, we will show that all of our approaches are applicable to various domain-specific text resources, even with limited training data.

This thesis does not cover the human information-seeking process as shown in Figure 1.1 in its entirety. In particular, we leave out a detailed discussion of the subprocesses (1) *intent detection* (human expresses information need to the machine) and (4) *interactive feedback loop* (human communicates degree of task completion). Instead, we focus on the MR component (2) and its ability to produce accurate results (3). Although studying user interaction is beyond the scope of this thesis, domain-specific information seeking requires real-world information needs and user interfaces. Therefore, we demonstrate the applicability of our MR components at the end of this thesis with a description of four prototypical implementations.

### 1.3 Research Objectives

This thesis addresses the problem of automatic understanding of unstructured text from domain-specific resources. The main hypothesis of this thesis is the following:

**Hypothesis.** Deep neural networks enable the efficient creation of general *Neural Machine Reading models* by capturing the distributional properties of text using *self-supervision*. MR models are able to process *domain-specific text resources* without expensive adaptation and with high error tolerance. MR enables end-to-end models to fulfill *task-specific information needs* with high accuracy by incorporating background knowledge, contextual information and task-specific training objectives.

We divide this general problem statement that we refer to as *Neural Machine Reading* into four subordinate research questions (RQ):

**RQ 1. What are general solutions to identify named entities in domain-specific text?** *Named entities are the smallest information units in classical IE and must be identified with high recall to facilitate effective downstream tasks. In domain-specific text, generic concepts, such as event and disease names, rare and emerging entities, such as new product and brand names must be recognized precisely. Recognition must not fail because of spelling errors, morphological variations or transliterations in the source text. A general model architecture must be efficiently applicable to a broad range of languages and domains, even with limited or zero training data. Furthermore, it is important to leverage contextual information from sentence, document, corpus and knowledge base to improve disambiguation compared to approaches based on local features.*

**RQ 2. How can Machine Reading models detect topics and structure in long documents?** *Long documents are often thoughtfully designed and structured by their authors to guide readers through the text. Established Information Extraction models, which operate on document or sentence level, are not taking this structure into account. To bridge the gap between IE and MR, a model must gain an understanding of global and local topics and handle topical shifts in long documents. A MR model must be able to identify coherent passages in the text and assign topic labels or section headings to each of the passages. MR models should be able to capture the context of entire documents, but operate coherently with word or sentence granularity.*

**RQ 3. How can we embed discourse structure into document representations?** *Neural models are often built upon distributed representations, which encode contextual information learned from large corpora of text. We aim to extend these representations with rich information about the discourse structure of a document. Specifically, we investigate if neural MR models benefit from contextualized document representations that embed information about entities and topical aspects. We raise the question if these representations can complement pre-trained word and sentence embeddings with missing information from long-range distance, such as introduction passages, coreferences, entity mentions, document titles and section headings. We expect from a document representation a complete and coherent semantic interpretation, comparable to a human reader who aims to understand the meaning of a text.*

**RQ 4. How effective are document representations for retrieving answer passages?** *Machine Reading models are designed to support human information-seeking tasks, e.g. retrieving answers from long documents, with high recall. We expect from a MR model that it does not rely on task-specific training data, but instead is able to form its beliefs based on unsupervised training data from generic sources, e.g. Wikipedia text, and distributional background knowledge. Specifically, we aim to understand how we can effectively apply the internal representations of a neural MR model, such as contextualized document embeddings, for nearest-neighbor search tasks. The representations must be utilizable in a wide range of tasks from similar domains, and should perform equally well as supervised ML models that are trained with explicit examples.*

In this thesis, we develop methods to meet all four RQs focused on three central tasks: Named Entity Linking, Topic Modeling and Answer Passage Retrieval.

## 1.4 Contributions

The main contribution of this thesis is the application of Neural Machine Reading to the problem of human information seeking across domain-specific text resources. We investigate this problem with respect to the research objectives posed in Section 1.3. The outcomes of our investigation are condensed into three main systems, which provide the following theoretical, practical and empirical contributions:

### **TASTY Named Entity Recognition and Linking** [Arnold et al., 2016b] (Chapter 3)

- We analyze design challenges and common errors for Named Entity Recognition (NER) and Disambiguation (NED) when applied to domain-specific text resources (Section 3.1).
- We compare distributed word embeddings and character-trigram based word encodings and show that trigrams are a robust and efficient representation for domain-specific text (Sections 3.2.1 and 3.1.3).
- We present the TASTY NER model based on trigram word encodings and Bidirectional Long Short-Term Memory (BLSTM) networks with state-of-the art NER performance on CoNLL2003 and five other English and German news datasets (Sections 3.2.2 and 3.3).
- We show that TASTY exceeds performance of other pretrained baselines in scenarios with small available training data, such as German car forum discussions and English biomedical text (Section 3.3).
- We present the TASTY NED model based on k-nearest neighbor search over entity embeddings and show that this neural model consistently scores high on four standard English datasets (Sections 3.2.3 and 3.3).
- We present the TASTY editor, an interactive application for Entity Linking as-you-type (Section 6.1).
- We present TraiNER, a process for bootstrapping domain-specific Named Entity Recognition using seed lists and active learning (Section 6.2).

### **SECTOR Topic Segmentation and Classification** [Arnold et al., 2019] (Chapter 4)

- We analyze challenges for Machine Reading and in particular show that structural topic information in long documents is not represented by current document representation models (Section 4.1).
- We introduce WIKISECTION, a dataset for the task of topic segmentation and classification in German and English for two specific domains: healthcare (diseases) and geopolitics (cities) (Section 4.1.2).



- We compare word-based and distributed sentence encodings and show that Bloom filters are an efficient sentence representation in a topic classification task (Section 4.3.1).
- We present SECTOR, a Neural Machine Reading architecture based on BLSTMs, which encodes structure and topic information much better than existing document embeddings (Sections 4.3.2 and 4.4).
- We propose to measure the deviation of SECTOR embeddings to identify topic shifts (Section 4.3.3).
- We show that SECTOR embeddings can be used to segment and classify passages into 25–30 domain-specific topics with high accuracy (Section 4.4).
- We present SMART-MD, an interface for clinical decision support based on the SECTOR Machine Reading architecture (Section 6.3).

#### **CDV Contextual Discourse Vectors for Answer Retrieval** [Arnold et al., 2020] (Chapter 5)

- We analyze challenges for MR-based Information Retrieval and identify task coverage, domain adaptability, contextual coherence and retrieval efficiency as important requirements (Section 5.1).
- We present CDV, a contextual discourse vector representation based on pre-trained language models and BLSTMs, which fulfills these requirements (Section 5.3).
- We propose to apply a multi-task CDV architecture for Answer Passage Retrieval based on entity and aspect embeddings (Section 5.2).
- We show that CDV can be trained with self-supervised data available from Wikipedia (Section 5.3.4).
- We adapt CDV for the healthcare domain and show that it is the most effective model when applied to three answer retrieval tasks without retraining (Section 5.4).
- We empirically analyze errors of the CDV model and propose how to address them in future work (Section 4.5).
- We present CDV Healthcare Answer Retrieval, a search interface for medical research papers based on the CDV Machine Reading architecture (Section 6.4).

To integrate our practical contributions, we open source all code of TASTY, SECTOR and CDV in a common Java framework *TeXoo*<sup>1</sup> released under Apache V2 license.

---

<sup>1</sup><https://github.com/sebastianarnold/TeXoo>

## 1.5 Thesis Outline

This thesis is structured around the vision of Machine Reading and three main systems that approach specific tasks for MR over domain-specific text resources. Here, we give an overview of each chapter:

**Chapter 1: Introduction.** We motivate the necessity for natural language understanding in the context of the human information-seeking process. We introduce the vision of Machine Reading to approach the challenges arising from high variance between specific languages, domains and tasks, and missing training data for domain-specific tasks. We divide this problem into five research questions, which we answer in the course of this thesis.

**Chapter 2: Background.** We discuss existing work from the literature that forms the groundwork for our automatic language understanding objective. We summarize the idea of distributed language representations, which are the fundamental background for our Neural Machine Reading approach. We further introduce specific methodical approaches for language-related sequence learning using deep neural networks.

**Chapter 3: A Robust Model for Efficient Entity Linking.** We identify design challenges and common errors specific to the task of Named Entity Recognition and Linking for domain-specific text. The main requirements of this task are robustness against spelling variations, effective consideration of contextual long-range information and efficient model creation from limited amounts of training examples. We introduce TASTY, an efficient sequence learning model for NER and NEL based on letter-trigram word encoding, Long Short-Term Memory and entity embeddings.

**Chapter 4: Coherent Topic Segmentation and Classification.** We broaden our view towards understanding document structure. We observe that documents often contain coherent passages that locally focus on a specific topic. We address the problem of identifying section boundaries in long documents and classifying each section into one of up to 30 domain-specific topics. We present SECTOR, a Neural Machine Reading architecture that learns a sentence-level latent topic embedding from training documents using Bidirectional Long Short-Term Memory and Bloom filter sentence encoding. We show that this embedding embodies all necessary information for the task of topic segmentation and classification.

**Chapter 5: Contextualized Document Representations for Answer Retrieval.** We build upon the SECTOR Neural Machine Reading model and extend it to cover named entities and entity-specific aspects. Our contextual discourse representation (CDV) can be trained with self-supervised data from Wikipedia and background knowledge from distributed entity and aspect embeddings. We apply the CDV model to human information-seeking tasks in the health-care domain over nine different English text resources without additional training data.

**Chapter 6: Systems.** Our models TASTY, SECTOR and CDV are based on a common code base TeXoo, which we release under an open source license. The models have been applied to various research prototypes and systems in industry. In this chapter, we present four applications which comprise the human information-seeking process as a whole.

**Chapter 7: Conclusion and Future Work.** We conclude this thesis with a summary and discussion of our contributions, and a reflection on our research objectives and limitations. Additionally, we present potential business applications and perspectives for Neural Machine Reading in future research.

## 1.6 Publications

This thesis is based on the following peer-reviewed articles (in the order of publication):

1. S. Arnold, F. A. Gers, T. Kiliyas, and A. Löser [2016b]. “Robust Named Entity Recognition in Idiosyncratic Domains”. In: *arXiv:1608.06757 [cs.CL]*<sup>2</sup>
2. S. Arnold, R. Dziuba, and A. Löser [2016a]. “TASTY: Interactive Entity Linking As-You-Type”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 111–115
3. R. Schneider, S. Arnold, T. Oberhauser, T. Klatt, T. Steffek, and A. Löser [2018]. “SmartMD: Neural Paragraph Retrieval of Medical Topics”. In: *The Web Conference 2018 Companion*. IW3C2, pp. 203–206
4. S. Arnold, R. Schneider, P. Cudré-Mauroux, F. A. Gers, and A. Löser [2019]. “SECTOR: A Neural Model for Coherent Topic Segmentation and Classification”. In: *Transactions of the Association for Computational Linguistics* 7, pp. 169–184
5. S. Arnold, B. van Aken, P. Grundmann, F. A. Gers, and A. Löser [2020]. “Learning Contextualized Document Representations for Healthcare Answer Retrieval”. In: *Proceedings of The Web Conference 2020*. ACM, pp. 1332–1343
6. J.-M. Papaioannou, S. Arnold, F. A. Gers, A. Löser, M. Mayrdorfer, and K. Budde [2020]. “Aspect-Based Passage Retrieval with Contextualized Discourse Vectors”. In: *[IN SUBMISSION] COLING 2020 System Demonstrations*<sup>3</sup>

The following articles have been published prior to this thesis. They describe classic approaches of symbolic Information Extraction and have led to the motivation of this work:

1. A. Löser, S. Arnold, and T. Fiehn [2012]. “The GoOLAP Fact Retrieval Framework”. In: *Business Intelligence*. Springer, pp. 84–97

<sup>2</sup>Due to many concurrent submissions on LSTM-based NER in 2016, we eventually published the final version of this article without peer-review. A good overview of concurrent work provides Lample et al. [2016].

<sup>3</sup>Paper is currently under review.

2. S. Arnold, D. Burke, T. Dörsch, B. Loeber, and A. Lommatzsch [2014]. “News Visualization Based on Semantic Knowledge.” In: *International Semantic Web Conference (Posters & Demos)*, pp. 5–8
3. U. Maqsud, S. Arnold, M. Hülfenhaus, and A. Akbik [2014]. “Nerdle: Topic-Specific Question Answering Using Wikia Seeds”. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pp. 81–85
4. S. Arnold, A. Löser, and T. Kiliyas [2015]. “Resolving Common Analytical Tasks in Text Databases”. In: *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP (DOLAP)*. ACM, pp. 75–84

## Chapter 2

# Background

In this chapter, we provide background information on the fundamental techniques relevant for this thesis. We begin in Section 2.1 by introducing the traditional approaches for Information Extraction based on discrete syntactic processing of text. In Section 2.2 we discuss the underlying idea of distributed language representations, such as the Vector Space Model, topic models, neural word embeddings and contextualized language models. In Section 2.3, we provide details on methods for sequential Machine Reading, such as finite-state machines, Recurrent Neural Networks, Long Short-Term Memory and Transformer architectures.

## 2.1 Information Extraction

Knowledge is often expressed by humans in form of written text, which is technically an unstructured representation that varies in style and format across different use cases. Typical text resources are news stories, chat messages, citations, technical documents or encyclopedia articles. *Information Extraction* (IE) describes the automatic transformation of these sources into structured or semi-structured representations, such as tuples, tables, lists, graphs or hierarchies. The main goal of this data integration step is to enable *Information Retrieval* (IR) using structured queries [Sarawagi, 2008].

Typically, IE is applied to short records with known boundaries e.g. citations, tweets or individual sentences. Additionally, many extraction tasks require to consider the context of associated paragraphs (e.g. headings, abstracts, snippets) or entire documents (e.g. news articles, discussion threads). To allow IE models to identify and process these elements, they often depend on *Natural Language Processing* (NLP) pipelines, which enrich plain text with linguistic or layout information using rule-based or statistical methods [Sarawagi, 2008]. In this section, we briefly describe classic approaches to IE which are based on this pipeline paradigm.

There exist a large variety of NLP pipeline implementations that come with pre-trained models for English language, such as Natural Language Toolkit (NLTK) [Loper and Bird, 2002], UIMA [Ferrucci and Lally, 2004], GATE [Cunningham et al., 2002], Stanford CoreNLP [Manning et al., 2014], Apache OpenNLP<sup>1</sup> or spaCy<sup>2</sup>. These pipelines have in common that they often rely on multiple independent models which are applied sequentially. This makes them

---

<sup>1</sup><https://opennlp.apache.org>

<sup>2</sup><https://spacy.io>

sensitive to variations and noise, and it is often not possible to adapt an entire pipeline to a different task or domain. The outcome is that most of these models do not meet our requirements to a Machine Reading model posed in Section 1.2. Next, we discuss the basic stages in the NLP pipeline in order to learn about their strengths and shortcomings.

### 2.1.1 Text Preprocessing

NLP models typically require a token-based input representation. The following three tasks are commonly applied in a preprocessing pipeline. Their aim is to transform the text given as one long character string into smaller normalized chunks that are later used for processing [Jurafsky and Martin, 2019]:

- **Tokenization** is the task of segmenting running text into words or word-piece tokens. Commonly used standards are the *Penn Treebank* standard released by the Linguistic Data Consortium [Marcus et al., 1993], or the word-piece tokens used in more recent Transformer models [Devlin et al., 2019]. A tokenizer needs to execute very fast and handle the ambiguity of words or special characters such as dots, apostrophes or hyphens. Therefore, the standard methods for tokenization use carefully-designed deterministic algorithms based on regular expressions and finite state automata. Tokenization is more complex in languages such as Chinese, where word boundaries are not usually expressed by spaces [Jurafsky and Martin, 2019].
- **Word normalization** is the task of transforming word tokens into a standard normal form. While feature-based models may improve from this reduction of information, recent neural models are often able to handle larger variances or even generalize better with non-normalized input. Typical practices for word normalization are case folding (mapping input to lower case) and lemmatization/stemming (normalizing words to their base form) [Jurafsky and Martin, 2019]. Furthermore, traditional NLP methods often apply stopword removal, which deletes the most common function words, such as articles and possessives before processing.
- **Sentence splitting** is the task of segmenting text into individual sentences. This step is often important, because many IE models are based on tokenized sentences as input. The most important cues for this task are punctuation (e.g. periods or question marks) and newline characters. Most models address sentence splitting and word tokenization jointly by deciding whether a period is part of a word or it is used as a sentence boundary marker. These methods are often based on deterministic models that include dictionaries of common abbreviations [Jurafsky and Martin, 2019].

The principle of word or word-piece tokenization and sentence splitting is not only used in NLP pipelines, but also in more powerful machine learning (ML) frameworks, such as PyTorch Transformers<sup>3</sup>. We will discuss token sequence based ML methods in Section 2.3.

---

<sup>3</sup><https://github.com/huggingface/transformers>

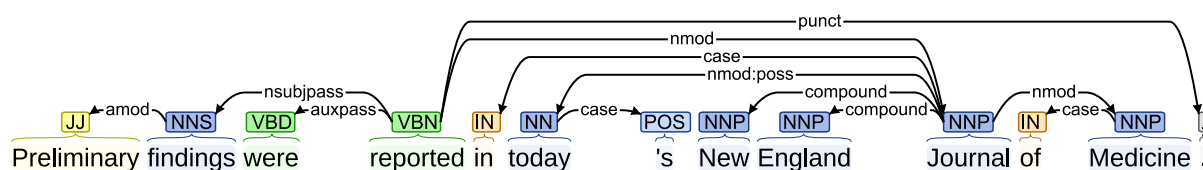


FIGURE 2.1: Part-of-speech and dependency parse result for the example sentence “Preliminary findings were reported in today’s New England Journal of Medicine.” generated by the Stanford CoreNLP framework.

### 2.1.2 Shallow Syntactic Parsing

A second stage in the NLP pipeline is often required to enrich the normalized tokens with linguistic information. For this task, a *shallow parse* can efficiently generate the necessary information from surface form features. Shallow models are based on syntax information and do not require full semantic understanding of language. The most important tasks are the following:

- **Part-of-speech tagging** (POS) is the task to assign each word a grammatical category from a fixed set for a specific language [Jurafsky and Martin, 2019]. The set of tags includes *closed class* types that contain function words with relatively fixed word memberships, such as determiners (DT), conjunctions (CC), prepositions (IN), pronouns and special character classes, such as parentheses or quotes. *Open class* types contain words which may continually be created or borrowed, such as proper nouns and names (NNP), common nouns (NN), verbs (VB), adjectives (JJ) or adverbs (RB). Typically, tags are further divided for plural nouns (NNS, NNPS) and verb inflections (VB\*). An important set of tags for English is contained in the 45-class Penn Treebank [Marcus et al., 1993]. POS taggers are often implemented using sequence learning algorithms (see Section 2.3).
- **Constituency parsing** is the task of grouping the words in a sentence into abstract phrase types, such as noun phrases, verb phrases or prepositional phrases [Sarawagi, 2008]. This operation is typically based on a parse tree that is generated by a *context-free grammar* over the POS tags of a sentence. The grammar rules can be learned from annotated *treebanks*. Often, constituency parse trees provide useful information for tasks such as entity or relation extraction.
- **Dependency parsing** is the task of identifying words in a sentence that form arguments of other words in the sentence [Sarawagi, 2008]. In contrast to constituency parsing, this task is based on a graph that describes the syntactic structure of a sentence in terms of the binary grammatical relations among its words, e.g. nominal subject, conjunction, direct or indirect object. Dependency parsers can be learned from large annotated dependency treebanks, such as Penn Treebank [Marcus et al., 1993], OntoNotes [Hovy et al., 2006] or Universal Dependencies [Nivre et al., 2016], which provides aligned annotations for multiple languages. Dependency trees provide useful information for many applications, such as coreference resolution, Question Answering or IE in general.

Figure 2.1 shows an example parse tree with POS and dependency annotations. This shallow information is often used as features in traditional symbolic IE models. However, recent ML approaches have replaced this extraction step by distributed representations which encode more complex relationships based on empirical linguistic knowledge. Distributed models are therefore less sensitive to noise and variance than symbolic representations. We will discuss these approaches in Section 2.2.

### 2.1.3 Deep Linguistic Processing

In addition to shallow syntactical parsing, many IE downstream tasks require a deeper semantic understanding of the text. This is necessary because propositions can be expressed using a large variety of words, which themselves are often ambiguous. In traditional NLP, this is done by further enriching the extractions with semantic annotations. Often, these tasks are not included in the preprocessing pipeline, but constitute a downstream task on their own. We briefly highlight the most important tasks:

- **Word sense disambiguation (WSD)** is the task to determine the sense of a word used in a particular context [Jurafsky and Martin, 2019]. Word senses are often defined as *glosses* in a dictionary, e.g. the word “mouse” can refer to “any small animal of various rodent and marsupial families” or “a palm-sized pointing device for a computer system”. The most commonly used resource for multi-lingual WSD is the WordNet lexical database [Fellbaum, 1998], which contains a large set of concept lemmas (nouns, verbs, adjectives and adverbs), each annotated with a set of senses. These senses are further clustered into *synsets*, which are sets of near-synonyms for a single concept. The goal of WSD is to assign a sense ID to each word in a text. WordNet has been further extended in BabelNet [Navigli and Ponzetto, 2012] with named entities from various sources, such as Wikipedia.
- **Semantic role labeling (SRL)** aims to capture the semantic relationships between a verb and noun arguments [Jurafsky and Martin, 2019]. For example, the verbs “sold”, “bought” or the noun “purchase” may refer to an acquisition event that holds roles like “agent” (the buyer), “theme” (the item to be bought) or “provider” (the seller). The task of SRL is to automatically find the semantic role of each predicate’s argument in a sentence. SRL algorithms often depend on features such as the governing predicate, phrase type of each constituent, headword of each constituent and the path in the parse tree from each constituent to the predicate. Although there is no universally agreed-upon set of roles, there are important resources, such as PropBank [Kingsbury and Palmer, 2002] and FrameNet [Baker et al., 1998] that hold large numbers of fine-grained roles that can be used to train and evaluate SRL models.
- **Coreference resolution** is the task to determine whether two mentions *corefer*, which means that they refer to the same *discourse entity* [Jurafsky and Martin, 2019]. The following example from Jurafsky and Martin [2019] shows a *coreference chain*, which is a set



of expressions that contains the initial entity mention (“Victoria”), pronouns (“she”, “her”) and other anaphora (“the 38-year-old”):

**Example 2.1.** [Victoria Chen]<sub>1</sub>, CFO of Megabucks Banking, saw [her]<sub>1</sub> pay jump to \$2.3 million, as [the 38-year-old]<sub>1</sub> became the company’s president. It is widely known that [she]<sub>1</sub> came to Megabucks from rival Lotsabucks.

An important property of referring expressions and their referents is that they must agree in number, person, gender or noun class. Coreference resolution is an important component of natural language understanding. It connects the symbolic token representations of the text with the mental *discourse model* [Karttunen, 1976] that a human reader builds incrementally when interpreting it. In some preprocessing pipelines, coreference information is resolved by replacing all referents with the initial mention.

Similar to shallow parsing, deep linguistic processing is today often replaced by distributional language modeling techniques. However, word senses, semantic roles and coreferences are still important latent features for language understanding. It remains an open challenge to include this information effectively in end-to-end models.

### 2.1.4 Named Entity Recognition

A central aspect of word sense disambiguation and coreference resolution tasks is that many mentions in a discourse refer to real-world instances of *named entities*, such as persons, companies or locations. The task of *Named Entity Recognition* (NER) is to identify the boundaries of named entity mentions in text and assign each mention a type. This task was originally introduced at MUC-6 [Grishman and Sundheim, 1996] and was later adapted in MUC-7 [Chinchor and Robinson, 1997] and the CoNLL-2003 shared task [Kim et al., 2004]. The objective is to classify each word in a document into one of four generic entity types, or none: PER (person names), LOC (locations), ORG (company and organization names) or MISC (miscellaneous). There exist similar schemes for domain-specific NER, e.g. gene, proteine, drug and disease names in the biomedical domain [Kim et al., 2003] or a large number of fine-grained types [Ling and Weld, 2012]. Ratnoff and Roth [2009] have illustrated the main challenges for this task based on the following example:

**Example 2.2.** SOCCER - [BLINKER]<sub>PER</sub> BAN LIFTED.  
[LONDON]<sub>LOC</sub> 1996-12-06 [Dutch]<sub>MISC</sub> forward had his indefinite suspension lifted by [FIFA]<sub>ORG</sub> on Friday and was set to make his [Sheffield Wednesday]<sub>ORG</sub> comeback against [Liverpool]<sub>ORG</sub> on Saturday. [Blinker]<sub>PER</sub> missed his club’s last two games after [FIFA]<sub>ORG</sub> slapped a world-wide ban on him for appearing to sign contracts for both [Wednesday]<sub>ORG</sub> and [Udinese]<sub>ORG</sub> while he was playing for [Feyenoord]<sub>ORG</sub>.

**Challenges for NER.** A NER model requires prior knowledge about entity names and type assignments (e.g. that “Udinese” is a soccer club). It has to make decisions based on non-local features such as context words (e.g. the overall topic “SOCCER”), linguistic patterns (e.g. “for

both  $[X]_{\text{ORG}}$  and  $[Y]_{\text{ORG}}$ ) or coreferences (e.g. “Wednesday” referring to “Sheffield Wednesday”). Additionally, noisy data (e.g. “BLINKER” is uppercased to mimic a bold headline) makes it hard to detect known entities.

**NER model evaluation.** NER models are evaluated by comparing their output with human annotations using micro-averaged precision, recall and  $F_1$  scores. These scores are calculated using true positives (TP), false positives (FP) and false negatives (FN) for *exact span match* between a set of predicted mentions  $\mathcal{P}_d$  and a set of relevant annotations  $\mathcal{R}_d$  mentioned in each document  $d$  of the dataset  $\mathcal{D}$  [Cornolti et al., 2013]:

$$\begin{aligned} \text{TP}_d &= |\{p \in \mathcal{P}_d \mid \exists r \in \mathcal{R}_d : \text{match}(p, r)\}| \\ \text{FP}_d &= |\{p \in \mathcal{P}_d \mid \nexists r \in \mathcal{R}_d : \text{match}(p, r)\}| \\ \text{FN}_d &= |\{r \in \mathcal{R}_d \mid \nexists p \in \mathcal{P}_d : \text{match}(r, p)\}| \end{aligned} \quad (2.1)$$

$$\begin{aligned} \text{Prec} &= \frac{\sum_{d \in \mathcal{D}} \text{TP}_d}{\sum_{d \in \mathcal{D}} (\text{TP}_d + \text{FP}_d)} \\ \text{Rec} &= \frac{\sum_{d \in \mathcal{D}} \text{TP}_d}{\sum_{d \in \mathcal{D}} (\text{TP}_d + \text{FN}_d)} \\ F_1 &= \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} \end{aligned} \quad (2.2)$$

**Rule-based approaches** utilize a set of hand-crafted rules to detect particular entity types. These rules are often based on lexical or syntactic features, linguistic patterns and domain knowledge. Some examples are word capitalization, regular expressions or shallow patterns based on POS tags. Often, these systems include domain-specific *gazetteers*, which are comprehensive lists of entity names that can be looked up. Traditional rule-based NER systems include FASTUS [Appelt et al., 1995], LaSIE-II [Humphreys et al., 1998], NetOwl [Krupka and Hausman, 1998], Facile [Black et al., 1998], SRA [Aone et al., 1998] or LTG [Mikheev et al., 1999]. These systems highly depend on precise rule definitions, therefore they are highly specific to a certain domain and often achieve high precision and low recall.

**Unsupervised learning approaches** focus on learning a set of rules by clustering unlabeled examples using a small set of seed rules [Collins and Singer, 1999], extraction patterns [Etzioni et al., 2005] or gazetteers [Nadeau et al., 2006]. These approaches can generalize well for different domains, but often depend on accurate linguistic preprocessing, which is challenging in domains with high variance.

**Supervised learning approaches** regard NER as multi-class classification or sequence labeling task. These methods require a large hand-tagged corpus which is used as training set to optimize the parameters of a ML model. The goal is that the model learns to generalize

over the training examples and discovers patterns and rules that can be applied to unseen examples. One critical step in supervised NER models is *feature engineering*, where text input is transformed into abstract representations. Typical word-based features are the current word, surface type (e.g. capitalized, all-capitalized, all-digits, alphanumeric, etc.), prefixes and suffixes, capitalization patterns, context words and previous label predictions [Ratinov and Roth, 2009]. Often, NER systems use additional features from linguistic preprocessing, such as POS tags, shallow parsing information and gazetteers. The training objective is most often to assign a tag to each word, such as BIOES (Begin, Inside, Outside, End, Singleton) of a named entity along with its type [Ratinov and Roth, 2009]. There also exist other tagging schemes with similar semantics, such as BIO2 or BILOU.

A large variety of ML algorithms has been applied to the NER task, such as Hidden Markov Models (HMM) [Rabiner, 1989; Bikel et al., 1997], Support Vector Machines (SVM) [Hearst et al., 1998; McNamee and Mayfield, 2002] or Decision Trees [Quinlan, 1986; Szarvas et al., 2006]. It was shown that sequence-based models such as Maximum Entropy Markov Models (MEMM) [McCallum et al., 2000] are most suitable to the NER task [Chieu and Ng, 2002; Bender et al., 2003; Curran and Clark, 2003]. Most prominently, Conditional Random Fields (CRF) [Lafferty et al., 2001] has become an early benchmark for NER [McCallum and Li, 2003; Krishnan and Manning, 2006].

**Deep-learning based approaches** have significantly improved the NER task [Li et al., 2020]. Pre-trained word embeddings such as word2vec [Mikolov et al., 2013a], GloVe [Pennington et al., 2014] and Fasttext [Bojanowski et al., 2017] improve sequence labeling tasks by enriching word representations with external knowledge learned from unlabeled text. Stacked encoder-decoder architectures of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) such as bidirectional Long Short-Term Memory (BLSTM) allow to include sentence-level and long-range context. For example, Collobert et al. [2011] apply CNN over the sequence of words in a sentence. Huang et al. [2015] and Lample et al. [2016] use a combination of BLSTM and CRF, Chiu and Nichols [2016] combine BLSTM and CNN. [Ma and Hovy, 2016] combine character-based word representations with a BLSTM-CNN-CRF model. Akbik et al. [2018] propose contextual string embeddings, which represent words by their stream of characters in a sequential context. Peters et al. [2018] introduce the ELMo embedding, which represent words using character-CNNs in a contextualized language model. Radford et al. [2018] and Devlin et al. [2019] extend the language model paradigm using pre-trained Transformers (GPT and BERT). These models are becoming a new paradigm of NER by replacing traditional word embeddings with powerful pre-trained language representations that can further be fine-tuned to a wide range of tasks [Li et al., 2020]. We will discuss the distributional properties leveraged by these models in more detail in Section 2.2 and provide detailed information on the individual sequence learning methods in Section 2.3.

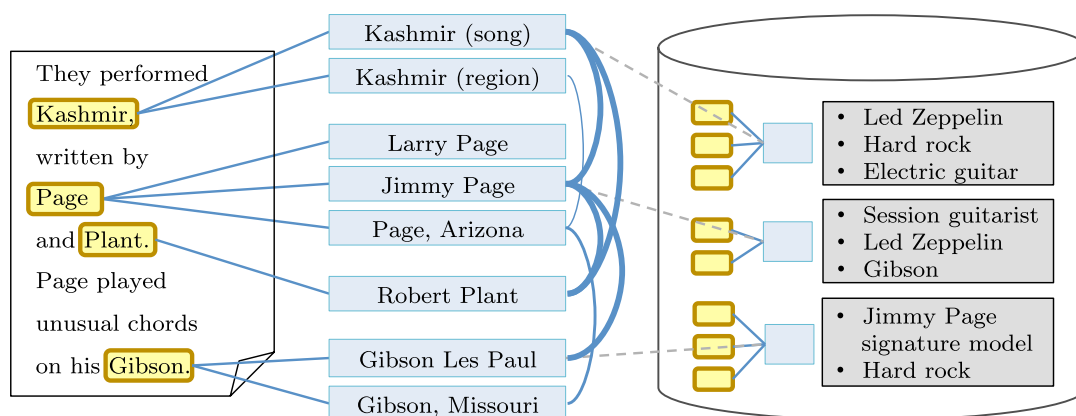


FIGURE 2.2: Example of a mention-entity graph used for NEL. Mentions in the text (left, highlighted in yellow) are linked to candidates (center). The goal is to assign the mention to the correct candidate based on the KB entry. The KB contains entities with relational and contextual information (right).

(Figure taken from Hoffart et al. [2011] CC BY-NC-SA 3.0)

### 2.1.5 Named Entity Linking

Many IE tasks require to explicitly associate named entity mentions with representations of objects in an *ontology*, such as a knowledge base (KB) or product database. The process of resolving named entities to a knowledge base is called *Named Entity Linking* (NEL) [Ji and Grishman, 2011]. This task is similar to the problem of word sense disambiguation, but here a domain-specific ontology is used instead of a complete lexical resource. A KB is typically focused on a specific subset of real-world entities (e.g. companies, products, or diseases) and it is potentially incomplete. Therefore, entity linking often includes the detection of *non-linkable* (NIL) entities or NIL clusters. NEL is typically approached in three separate stages: mention recognition (*extraction*, NER), candidate generation (*search*) and candidate ranking (*disambiguation*, NED) [Hachey et al., 2013]. In this Section, we briefly discuss the candidate generation and disambiguation processes.

**Knowledge Bases.** Since most tasks require to cover a large number of real-world entities, the most common ontology used for NEL is Wikipedia. Wikipedia not only provides unstructured textual information about over 6 million entities and concepts in many languages, it also contains many useful structured elements such as page titles, redirects, hyperlink anchor texts, disambiguation pages, infoboxes and categories [Hachey et al., 2013]. Some KBs that contain structured data derived from Wikipedia are Wikidata [Vrandečić and Krötzsch, 2014], Freebase [Tanon et al., 2016], YAGO2 [Hoffart et al., 2013] or DBpedia [Lehmann et al., 2015]. A popular evaluation task for NEL is TAC Knowledge Base Population (KBP) [Ji and Grishman, 2011].

**Methods for candidate generation.** As a first step, a set of candidates from the KB is generated for every mention. An ideal candidate generation should balance precision and recall to capture the correct entity while reducing the amount of computation for disambiguation

[Hachey et al., 2013]. Typically, this involves matching entity names using dictionary-based techniques such as exact or partial match [Cucerzan, 2007; Varma et al., 2009; Ratnov et al., 2011]. It is common practice to build name dictionaries by gathering aliases and synonyms from Wikipedia pages, hyperlinks and metadata [Bunescu and Pasca, 2006; Mihalcea and Csomai, 2007]. Furthermore, surface form expansion for acronyms [Varma et al., 2009; Zhang et al., 2011] can be used to generate more synonyms. Spelling correction techniques [Chen et al., 2010; Zheng et al., 2010; Shen et al., 2012] are often used to normalize candidate queries. In addition, Web search results [Han and Zhao, 2009; Dredze et al., 2010; Cheng et al., 2011] and query click logs [Chakrabarti et al., 2012; Taneva et al., 2013] can be utilized to generate candidates with higher variance.

**Features for candidate ranking.** In the second step, the candidates are ranked and a decision is made to assign an entity ID (or NIL) to each mention. Ideally, the method should rank all mentions in a document collectively to obtain a coherent assignment. Typical context-independent ranking features include name string comparison (e.g. using edit or hamming distance), entity type and popularity information [Shen et al., 2015]. However, in most cases it is crucial to include context-dependent features, which are often extracted from the interaction between mention context and entity descriptions. Typical examples are bag-of-words, feature vectors, language and topic modeling and coherence between entity mappings [Shen et al., 2015].

**Supervised methods for candidate ranking.** A simple supervised approach to the candidate ranking problem is to apply *binary classification* to decide if an entity mention refers to a candidate entity. To rank a larger number of candidates and handle unbalanced training data, many NEL systems utilize *learning to rank* [Herbrich, 2000]. This framework takes the relations between candidate entities into account instead of considering them as independent. Bunescu and Pasca [2006] utilize an SVM ranking model over link anchor context and categories from Wikipedia. Cucerzan [2007] use a similar approach, but rank candidates using the scalar product of a document feature vector with all candidate feature vectors. Varma et al. [2009] rank candidates based on cosine similarity between the mention paragraph and the text of the candidate page. Zheng et al. [2010] utilize a ranking perceptron algorithm to learn pairwise ranking. Zhang et al. [2011] utilize a topic model to generate additional semantic features for ranking. Ratnov et al. [2011] distinguish between local and global disambiguation techniques. They propose to use normalized Google distance and pointwise mutual information (PMI) as global relatedness measures to achieve more coherent disambiguations.

**Graph-based approaches** make use of relationships that are explicitly modeled in a KB or can be inferred from Wikipedia page structure. Hoffart et al. [2011] propose a robust and coherent method for collective disambiguation by applying a dense subgraph algorithm over a mention–entity graph (a visualization is shown in Figure 2.2). Shen et al. [2012] propose a semantic network similarity measure based on Wikipedia concepts and hyperlink structure.

Moro et al. [2014] approach NEL and WSD tasks jointly by applying densest subgraph heuristics to a semantic interpretation graph. Durrett and Klein [2014] use a factor graph to associate mentions with coreference, entity type and link variables using CRF and minimum Bayes risk decoding.

**Neural architectures** allow to replace labor-intensive feature engineering with learned dense representations of words, sentences, phrases or documents that can capture a larger amount of context. Typically, neural models learn a similarity function between a mention and each entity candidate using labeled examples for supervised training. Mentions are mostly represented by their characters and surrounding context. Entity candidates are often represented using descriptions and type information derived from the KB.

He et al. [2013] apply feed-forward networks to learn this similarity function using bag-of-words context representations. The representations are pre-trained using stacked denoising autoencoders and then fine-tuned with labeled examples. Sun et al. [2015] and Francis-Landau et al. [2016] utilize CNNs to learn the similarity function based on semantic representations of mention and entity contexts. Sil et al. [2018] extend the semantic representation using multi-lingual word embeddings and CNN sentence encoding. They encode left and right context using LSTM and Neural Tensor Networks. Pappu et al. [2017] propose an efficient entity disambiguation by extending the Paragraph Vectors model [Le and Mikolov, 2014]. This model encodes an entity embedding jointly based on global token context and context from surrounding entities.

Gupta et al. [2017] use a modular model which encodes an entity embedding using descriptions, mention contexts and fine-grained types using LSTM and Feed-forward (FF) networks. Gillick et al. [2019] combine two network structures into a dual encoder model: One network encodes mentions including their contexts, the other one encodes entities from the KB. A key property of this model is that it does not require interaction between both encoders, therefore efficient retrieval without candidate generation is possible. Logeswaran et al. [2019] utilize pre-trained Transformers for matching candidate and mention contexts. They show that domain adaptive pre-training allows the model to be applied in a *zero-shot task*, where mentions are linked to unseen entities without in-domain labeled examples.

In summary, we have seen that neural NEL models are increasingly moving towards learning a universal linking function between unstructured text and structured world knowledge. This objective refers back to WSD and SRL, as it often includes general concepts, unseen entities, topics and, ideally, coreferences. At the same time *Information Retrieval*, which is traditionally considered a downstream task to structured Information Extraction, is now often directly applied to neural representations [Mitra and Craswell, 2018; Onal et al., 2018]. It remains the question if the intermediate symbolic representations produced by Entity Linking are still required for neural MR? In the next section, we will discuss a complementary perspective, which aims towards understanding language generically based on distributed representations.

## 2.2 Distributed Language Representations

Natural language is an inherent representation of human communication and knowledge. It can be perceived as a sequence of discrete symbols, e.g. words, phonemes or sounds, that follows rules that both the speaker and the hearer know [Chomsky, 1965]. The idea of *distributional semantics* is to transform these linguistic elements into generalized representations that encode meaning rather than discrete syntax [Turney and Pantel, 2010]. One central property of distributed representations is *semantic similarity*, which describes a measurable relation of nearness between the meaning of two elements. In general, semantic similarity is closely related to the phenomena of *synonymy* (two words have the same meaning), *hyponymy* (one word is a generalization or specialization of another word) or *antonymy* (two words have the opposite meaning). Symbolic language representations are limited when words or phrases are ambiguous, as in the case of *polysemy* (a single word can be used to express different meanings) and *homonymy* (two words with same sound or spelling have different meanings). Therefore distributional models aim to capture the meaning of linguistic elements based on the context they appear in.

### 2.2.1 Distributional Hypothesis

Harris [1954] describes how the distributional structure of language is characterized by the relations between its elements. His hypothesis assumes that “these relations really hold in the data investigated” [1954, p. 149]. They can be empirically observed by frequency and relative position between elements and basic classes of co-occurring elements. However, Harris states that “the distinction between distributional structure and meaning is not yet always clear” [1954, p. 151]. This means that a morpheme or word has no single or central meaning, or even “a continuous or coherent range of meanings” [1954, p. 152].

This mismatch between symbols and their semantic interpretation is picked up by Firth [1957], who manifests that a word is characterized “by the company it keeps” and its sense can therefore be determined by its context. Sahlgren [2008] solidifies this empirical hypothesis with a theoretical foundation. Specifically, he points out two relationships of linguistic context based on structuralist theory [de Saussure, 1916]: *Syntagmatic relations* concern positioning and hold elements that co-occur in sequential combinations in a text. *Paradigmatic relations* concern substitution and hold elements that occur in the same context but not at the same time (see Figure 2.3). This differential view on meaning is the foundation of the *refined distributional hypothesis* [Sahlgren, 2008, p. 7]:

*A distributional model accumulated from co-occurrence information contains syntagmatic relations between words, while a distributional model accumulated from information about shared neighbors contains paradigmatic relations between words.*

In the following sections, we summarize existing representation models based on this hypothesis. We will show that the distributional hypothesis enables the automatic construction of neural language models, which constitute the foundation for Neural Machine Reading.

Syntagmatic relations Combinations: “ $x$ and $y$ and...”	Paradigmatic relations Selections: “ $x$ or $y$ or...”			
	she	adores	green	paint
	he	likes	blue	dye
	they	love	red	colour

FIGURE 2.3: Examples for syntagmatic and paradigmatic relations.  
(Figure taken from Sahlgren [2008])

## 2.2.2 Semantic Vector Space Model

The idea of the *Vector Space Model* (VSM) is to represent linguistic elements—traditionally documents in a collection—as a point in space [Salton et al., 1975]. In this space, points that are close together are semantically similar and points that are distant are semantically different. Following the distributional hypothesis, such a model can automatically be built from a corpus of text using algorithms that utilize the distribution of terms in that corpus. Therefore the VSM constitutes a suitable foundation for self-supervised Machine Reading systems.

**Bag-of-words vector space model.** The original bag-of-words vector space model (BOW) is represented by a *term–document matrix*  $\mathbf{X}$  [Salton et al., 1975]. In this matrix, the rows correspond to terms (e.g. words) and the columns to documents. For a corpus of  $n$  documents  $D = d_{1\dots n}$  with  $m$  unique terms  $w_{1\dots m}$ , the matrix  $\mathbf{X}$  will have  $m$  rows and  $n$  columns. We assign each element  $x_{ij}$  the frequency of the  $i$ -th term  $w_i$  in the  $j$ -th document  $d_j$ . Often, a normalized frequency measure such as *term frequency – inverse document frequency* (TF-IDF) [Jones, 1972] is used to give higher weight to discriminative words:

$$\begin{aligned}
 \text{TF}_{w,d} &= \log(1 + |\{w' : w' = w \wedge w' \in d\}|) \\
 \text{IDF}_{w,D} &= \log\left(1 + \frac{n}{|\{d \in D : w \in d\}|}\right) \\
 \text{TF-IDF}_{w,d,D} &= \text{TF}_{w,d} \cdot \text{IDF}_{w,D}
 \end{aligned} \tag{2.3}$$

As most documents contain only a small fraction of the entire vocabulary,  $\mathbf{X}$  is a sparse matrix, i.e. most of its entries are 0. We can now use the  $j$ -th column vector as *bag-of-words* vector representation of the  $j$ -th document:

$$\mathbf{x}_{\text{bow}}(d_j, D) = \mathbf{x}_{\text{bow}}(\mathbf{X}, j) = \mathbf{x}_{:j} = \sum_{w_i \in d_j} \mathbf{e}_{w_i} \tag{2.4}$$

where  $\mathbf{e}_i \in \{0, 1\}^m$  is a *one-hot* representation of a word  $w_i$  by the  $i$ -th unit vector in  $\mathbb{R}^m$ . This *local distributed representation* captures (to some degree) the meaning of a document, although the sequential order of terms is lost [Turney and Pantel, 2010]. Nevertheless, it captures an important aspect of semantics and is often used in IR to calculate the relevance of a document to a query [Salton et al., 1975].



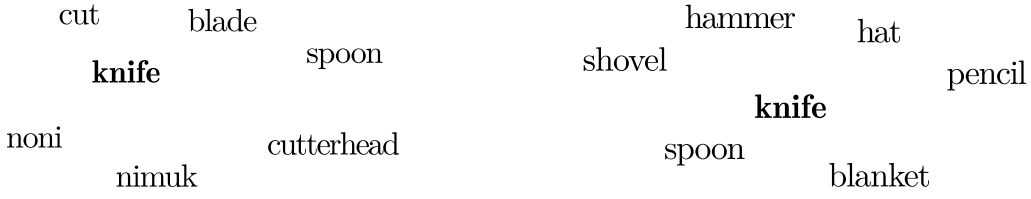


FIGURE 2.4: Syntagmatic (left) and paradigmatic (right) neighborhood examples of the word “knife”. (Figures taken from Sahlgren [2008])

**Probabilistic topic models.** One important application of term–document frequency distributions is *probabilistic topic modeling* [Blei, 2012]. Topic modeling algorithms aim to discover and organize latent themes discussed in documents. The most prominent algorithm, *latent Dirichlet allocation* (LDA) [Blei et al., 2003] uses a generative process to assign words to topics and topics to documents. A topic is here defined as a distribution over words of a fixed vocabulary, and each document holds a distribution over a fixed number of topics. For example the “genetics” topic assigns high probability to words referring to genes, DNA and heredity in general. A topic model will therefore assign the “genetics” topic with high probability to documents containing these words. One significant drawback of this model is that it is entirely based on BOW over the global document context, i.e. it ignores word order and it does not respect local word contexts. Therefore, an important step towards semantic models of language is to represent words in their local context.

**Vector space models based on syntagmatic associates.** When investigating the similarity between individual words or short phrases, the BOW model is very limited. The reason is that BOW vectors of two documents without common words are always orthogonal. Therefore, Deerwester et al. [1990] shift the focus to analyzing single terms by looking at their syntagmatic associates. These relations are represented by the row vectors in the term–document matrix instead of column vectors. They introduce the *Latent Semantic Analysis* (LSA) algorithm, which applies singular-value decomposition (SVD) to decompose  $\mathbf{X}$  into  $k \in \{50, \dots, 1500\}$  orthogonal factors [Landauer and Dumais, 1997]. Words can be translated to vectors in the *semantic space* spanned by the approximated factor weights. They use truncated SVD,  $\hat{\mathbf{X}} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$  to transform the  $i$ -th row vector of  $\mathbf{X}$  into the LSA representation of the  $i$ -th term [Turney and Pantel, 2010]:

$$\mathbf{x}_{\text{lsa}}(w_i, D) = \mathbf{x}_{\text{lsa}}(\mathbf{X}, \mathbf{\Sigma}_k, \mathbf{V}_k, i) = \mathbf{\Sigma}_k^{-1} \mathbf{V}_k^\top \mathbf{x}_i \quad (2.5)$$

More generally, distributional word meaning can be inferred from *word–context matrices*, where context is a text region covering a phrase, sentence, paragraph, chapter or document [Turney and Pantel, 2010]. The semantic properties of the resulting vector space not only depend on the length of the context, but also on the type of relation used to build the model. Because the LSA algorithm is based on syntagmatic relations, it encodes word associations rather than taxonomic similarities [Schütze and Pedersen, 1993]. For example, Figure 2.4 (left) demonstrates the syntagmatic neighborhood for the word “knife” based on a 10M word corpus

of English high-school level texts and a section-level context of roughly 150 words [Sahlgren, 2008]. “Noni” and “Nimuk” are person names related to a story where a knife plays a role. These names are therefore *semantically associated* to “knife” but not *semantically similar* [Turney and Pantel, 2010].

**Vector space models based on paradigmatic parallels.** A different perspective on distributional word context is to look at close neighbors of each term. This is achieved by collecting text data in a *words-by-words co-occurrence matrix* that holds frequencies of words occurring together within a context window. In a directional co-occurrence matrix, rows and columns correspond with left and right context. In symmetric co-occurrence matrices, frequencies are calculated from the entire window, and therefore rows and columns are equal [Sahlgren, 2008]. The *Hyperspace Analogue to Language* (HAL) model [Lund and Burgess, 1996] uses a paradigmatic approach to encode distributional word context. The HAL model uses a directional matrix over a sliding window of 5 words to the left and right as context [Lund and Burgess, 1996]. For example, Figure 2.4 (right) demonstrates the paradigmatic neighborhood for the word “knife” with a small 2+2 words context on the same corpus used for the first example. While “spoon” occurs in both syntagmatic and paradigmatic context, all the words in this figure intuitively have a higher *taxonomical similarity* than the words in the previous example [Turney and Pantel, 2010]: they are singular neuter nouns from the area of household items and most of them share a common hyponym “tool”.

**Similarity measures.** The most popular measure for semantic nearness captures the idea that the angle between two vectors in semantic space  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$  reflects the similarity between them. We calculate this measure using *cosine similarity* [Turney and Pantel, 2010]:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\sqrt{\mathbf{a} \cdot \mathbf{a}} \cdot \sqrt{\mathbf{b} \cdot \mathbf{b}}} = \frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} \quad (2.6)$$

If  $\mathbf{a}$  and  $\mathbf{b}$  are normalized to unit length, cosine similarity equals the dot product. The value ranges from -1 (vectors are pointing in opposite direction) over 0 (vectors are orthogonal) to 1 (vectors point in the same direction) [Turney and Pantel, 2010]. In practice, cosine similarity is often applied to positive vectors (e.g. term frequency vectors), where a value of 1 depicts the highest similarity, while a value of 0 means they are uncorrelated.

Other popular distance measures include the geometric measures *Euclidean distance* and *Manhattan distance* as well as information theory measures *Kullback-Leibler*, *Dice coefficient* and *Jaccard coefficient* [Bullinaria and Levy, 2007; Manning et al., 2008].

### 2.2.3 Neural Distributed Language Representations

Statistical language modeling based on global matrix factorization, such as LSA, is problematic when applied to large data sets with high variance. Not only it is computationally expensive to handle large sparse matrices. More importantly, in order to obtain generalization,

it is necessary to handle longer context sequences and a large number of infrequent words or phrases. Therefore, neural models aim to approximate a *language model* (LM) by observing a large number of samples in context. They often learn a mathematical *embedding* from the high-dimensional word representation to a continuous vector space with much lower dimension.

**Neural probabilistic language model.** Bengio et al. [2003] propose a *neural probabilistic language model* that utilizes neural networks to learn a distributed representation of words along with a probability function for word sequences. The main idea is to represent language by the conditional probability of the next word given all previous words:

$$p(w_1, \dots, w_T) = \prod_{t=1}^T p(w_t \mid w_1, \dots, w_{t-1}) \quad (2.7)$$

where  $w_1, \dots, w_T$  is a sequence of words. The complexity of this model can be considerably reduced using the *n-gram* model, which reduces the context to combinations of the last  $n - 1$  words:

$$p(w_t \mid w_1, \dots, w_{t-1}) \approx p(w_t \mid w_{t-1}, \dots, w_{t-n+1}) \quad (2.8)$$

This probability distribution can be computed by a neural network with a *softmax* output layer to produce a vector with positive probabilities for a word  $w_t$  summing to 1:

$$p(w_t \mid w_{t-1}, \dots, w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \quad (2.9)$$

They use a feed-forward neural network with word-feature mapping weights  $\mathbf{W}_x$ , hidden layer weights  $\mathbf{W}_h$ , output layer weights  $\mathbf{W}_y$  and corresponding bias terms  $\mathbf{b}_x, \mathbf{b}_h$ :

$$\mathbf{y} = \mathbf{W}_x \mathbf{x} + \mathbf{b}_x + \mathbf{W}_y \tanh(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h) \quad (2.10)$$

Input features  $\mathbf{x}$  are a concatenation of all context words. The network is trained with *stochastic gradient descent* (SGD) and a loss function that maximizes the penalized log-likelihood for the training corpus. The advantage of this neural model over traditional term-frequency models is a more compact and smoother representation that better preserves linear regularities among words. In addition, a neural model can be trained with a larger number of conditioning variables, because it scales linearly, not exponentially with the number of variables [Bengio et al., 2003].

**Word embeddings based on local context windows.** Mikolov et al. [2013a] propose two model architectures to learn accurate distributed word representations with minimized computational complexity. The *Continuous Bag-of-Words model* (CBOW) predicts the current word based on its context, while the *Skip-gram model* predicts surrounding words for a given word (see Figure 2.5). By learning word vectors without constructing a full language model, this approach improves efficiency for training with larger datasets. Especially the Skip-gram model, often

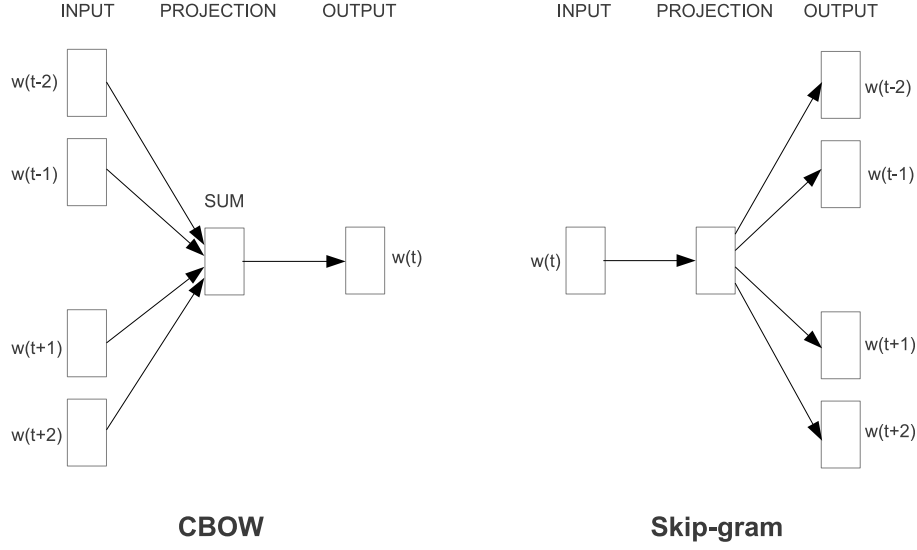


FIGURE 2.5: Architecture of CBOW (left) and Skip-gram (right) models.  
(Figure taken from Mikolov et al. [2013a])

referred to as *word2vec*<sup>4</sup>, has proven to be an efficient and accurate model. The objective of this model is to encode paradigmatic relations for a given word by maximizing the log-likelihood for all surrounding words  $\mathcal{C}_t$  in a window of typically 4-10 words [Mikolov et al., 2013b]:

$$\mathcal{L}_{\text{skip-gram}} = \sum_{t=1}^T \sum_{c \in \mathcal{C}_t} \log p(w_c | w_t) \quad (2.11)$$

Word2vec uses a similar neural architecture as Bengio et al. [2003] with a single hidden layer. Instead of softmax, they use *negative sampling*, a simplified form of Noise Contrastive Estimation [Gutmann and Hyvärinen, 2012], to reduce the training complexity of the model. Here, the objective is to maximize the negative log-likelihood:

$$\mathcal{L}_{\text{word2vec}} = \sum_{t=1}^T \left( \sum_{c \in \mathcal{C}_t} \log(1 + e^{-s(w_t, w_c)}) + \sum_{n \in \mathcal{N}_{t,c}} \log(1 + e^{-s(w_t, w_n)}) \right) \quad (2.12)$$

where  $\mathcal{N}_{t,c}$  is a set of negative word examples and  $s(w_t, w_c) = \mathbf{u}_{w_t}^\top \mathbf{v}_{w_c}$  is a scoring function using the scalar product between word vector  $\mathbf{u}_{w_t}$  and context vector  $\mathbf{v}_{w_c}$  [Pennington et al., 2014]. Additionally, they observed that subsampling frequent words during training results in better representations of uncommon words [Mikolov et al., 2013b].

**Character-based word embeddings.** Word-based embedding models have the disadvantage, that their one-hot input representation relies on a fixed vocabulary, and it is therefore not possible to generate word vectors for out-of-vocabulary (OOV) words. Furthermore, such representations ignore the morphology of words and will assign distinct vectors to different word

<sup>4</sup><https://code.google.com/p/word2vec>

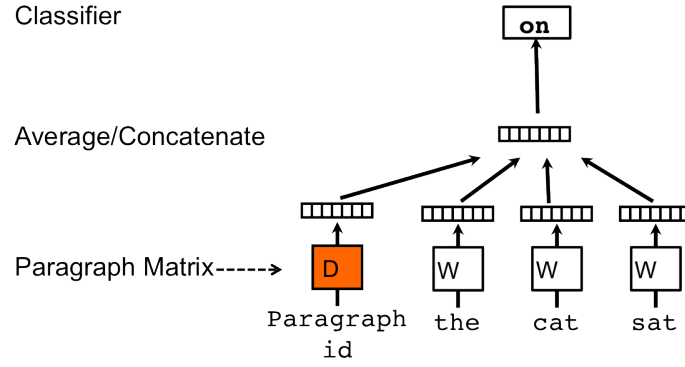


FIGURE 2.6: Architecture of the Paragraph Vector framework.  
(Figure taken from Le and Mikolov [2014])

forms or writings. Additionally, the accuracy of infrequent words, e.g. in morphologically rich languages or specialized domains, may suffer from too few training examples. To address this issue, Bojanowski et al. [2017] propose *Fasttext*<sup>5</sup>, a word embedding approach that extends the Skip-gram model with character-based word representations similar to the approach of Schütze [1993]. In this model, each word is represented as a bag of *character n-grams* with  $3 \leq n \leq 6$ . For example, the word “where” is represented by a special sequence `<where>` which yields the following 3-grams: `<wh, whe, her, ere, re>`. The model efficiently learns a representation for each n-gram and represents a word as the sum of n-gram vectors. To achieve this, Eq. 2.12 is extended with a different scoring function:

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c \quad (2.13)$$

where  $\mathcal{G}_w$  is the set of n-grams appearing in  $w$  and  $\mathbf{z}_g$  is the vector representation of n-gram  $g$ .

**Neural word embeddings based on global word co-occurrences.** Pennington et al. [2014] propose the *Global Vectors for word representation* (GloVe)<sup>6</sup> model. In contrast to word2vec, which utilizes local word context windows, GloVe is based on a global word–word co-occurrence matrix and therefore is able to leverage a larger amount of context from syntagmatic relations. To increase efficiency, GloVe is trained only on the nonzero elements of the matrix using a specific weighted least squares model. They use a similar log-bilinear regression objective as Mikolov et al. [2013a]. This method achieves higher accuracy than word2vec for models based on the same amount of training data and computation time during training.

**Distributed representations of sentences and documents.** Most of the previous models focus on the representation of single words. Aiming at the representation of longer phrases, it was shown that the Skip-gram model exhibits a linear structure that allows accurate *composition* of words [Mikolov et al., 2013b; Mitchell and Lapata, 2010]. This is possible by using simple

<sup>5</sup><https://fasttext.cc>

<sup>6</sup><https://nlp.stanford.edu/projects/glove/>

vector arithmetics, especially element-wise addition or weighted average of word vectors, to create fixed-length representations from multiple words. However, similar to the bag-of-words approach, the main drawback from this method is that word order information is lost during this process. This is especially problematic for the representation of long documents.

Therefore, Le and Mikolov [2014] propose an unsupervised algorithm that addresses these issues for texts of variable length. Their *Paragraph Vectors* (ParVec) model extends the CBOW model by optimizing multiple shared word vectors in combination with one unique paragraph vector per text. This process is visualized in Figure 2.6: words are represented by columns in matrix  $\mathbf{W}$ , and paragraphs are mapped to columns in matrix  $\mathbf{D}$ . At training time, these matrices are optimized using a neural network:

$$\mathbf{y} = \mathbf{U}h(w_{t-k}, \dots, w_{t+k}; \mathbf{W}, \mathbf{D}) + \mathbf{b} \quad (2.14)$$

where  $h$  is constructed by concatenation or average of word vectors and  $\mathbf{U}, \mathbf{b}$  are parameters of a hierarchical softmax objective encoded as binary Huffman tree. At prediction time, an inference step is performed through the word matrix  $\mathbf{W}$  of the network. Finally, a paragraph representation is computed by performing SGD through the paragraph matrix  $\mathbf{D}$ .

## 2.2.4 Contextualized Language Models

The pre-trained word representation models discussed in the previous section have become key components for neural language understanding. However, most of them are focused on paradigmatic relations and ignore how words are used differently in different linguistic contexts, such as polysemous words. The goal of contextualized language models is to capture context-dependent aspects of word meaning while maintaining the effectiveness of unsupervised language representations.

**Embeddings from Language Models.** To tackle this challenge, Peters et al. [2018] propose *Embeddings from Language Models* (ELMo), a deep contextualized word representation using a recurrent neural network (RNN) with higher complexity. More specifically, they extend the idea of Bengio et al. [2003] by learning a *bidirectional language model* with multiple layers of internal representations. The overall objective of this LM is to maximize the log likelihood of predicting each word in a document  $d = (w_1, \dots, w_T)$  given its left and right context:

$$\begin{aligned} \mathcal{L}_{\text{ELMo}} = \sum_{t=1}^T & \left( \log p(w_t \mid w_1, \dots, w_{t-1}; \Theta_{\text{CNN}}, \vec{\Theta}_{\text{LSTM}}, \Theta_s) \right. \\ & \left. + \log p(w_t \mid w_{t+1}, \dots, w_T; \Theta_{\text{CNN}}, \vec{\Theta}_{\text{LSTM}}, \Theta_s) \right) \end{aligned} \quad (2.15)$$

Internally, the parameters  $\Theta$  refer to three different types of network layers, which are jointly optimized: Input word vectors  $\mathbf{x}_t$  are encoded on character-level using a convolutional neural network (CNN) [LeCun et al., 1989] with parameters  $\Theta_{\text{CNN}}$ . Internal hidden states  $\mathbf{h}_{t,k}$  are

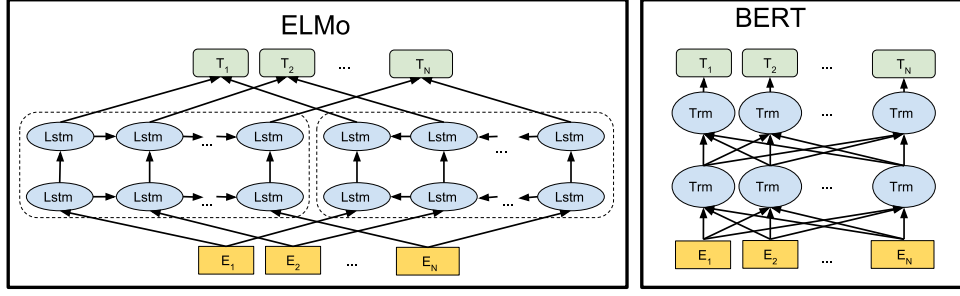


FIGURE 2.7: Model architectures of ELMo (left) and BERT (right).  
(Figure adapted from Devlin et al. [2019] CC BY 4.0)

computed by  $L$  interconnected layers of Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] for forward and backward direction  $\vec{\Theta}_{\text{LSTM}}, \tilde{\Theta}_{\text{LSTM}}$ . The output objective  $y_t$  is calculated by a softmax layer. The first layer uses the word vectors as input:

$$\begin{aligned}
 \mathbf{h}_{t,0} &= \mathbf{x}_t = \text{CNN}(w_t, \Theta_{\text{CNN}}) \\
 \vec{\mathbf{h}}_{t,k} &= \text{LSTM}(\mathbf{h}_{t,k-1}, \vec{\mathbf{h}}_{t-1,k}; \vec{\Theta}_{\text{LSTM}}) \\
 \tilde{\mathbf{h}}_{t,k} &= \text{LSTM}(\mathbf{h}_{t,k-1}, \tilde{\mathbf{h}}_{t+1,k}; \tilde{\Theta}_{\text{LSTM}}) \\
 \mathbf{h}_{t,k} &= \vec{\mathbf{h}}_{t,k} \oplus \tilde{\mathbf{h}}_{t,k} \\
 y_t &= \text{softmax}(\mathbf{h}_{t,L})
 \end{aligned} \tag{2.16}$$

where  $\oplus$  depicts vector concatenation and  $k \in 1, \dots, L$  is the layer index. Finally, the representation is calculated by collapsing all layers into a single vector:

$$\mathbf{x}_{\text{elmo}}(w_t, d) = \gamma \sum_{k=1}^L s_k \mathbf{h}_{t,k} \tag{2.17}$$

where  $\gamma$  is a scaling parameter and  $s$  are softmax-normalized layer weights learned for a specific task. It was shown that ELMo representations capture polysemous word meanings from context and therefore significantly improve many word-based tasks compared to GloVe.

**Bidirectional masked language models.** One inherent problem of RNN-based LMs is that the recurrent architecture makes it hard to maintain long-range dependencies. In these networks, the sequential information has to travel a long path through the cells, leading to the *vanishing gradient* problem [Pascanu et al., 2013]. Although LSTMs can partly solve this issue using gating mechanisms, their architecture becomes very complex, because every time step in a sequence depends on all preceding time steps (see Eq. 2.7). This leads to high memory bandwidth and prevents the parallelization of computations. An additional problem is that RNN-based LMs handle left and right context of a word as two independent sequences (see Eq. 2.15 and Figure 2.7). Combining both directions into a true bidirectional model would exponentially increase the number of dependent weights.

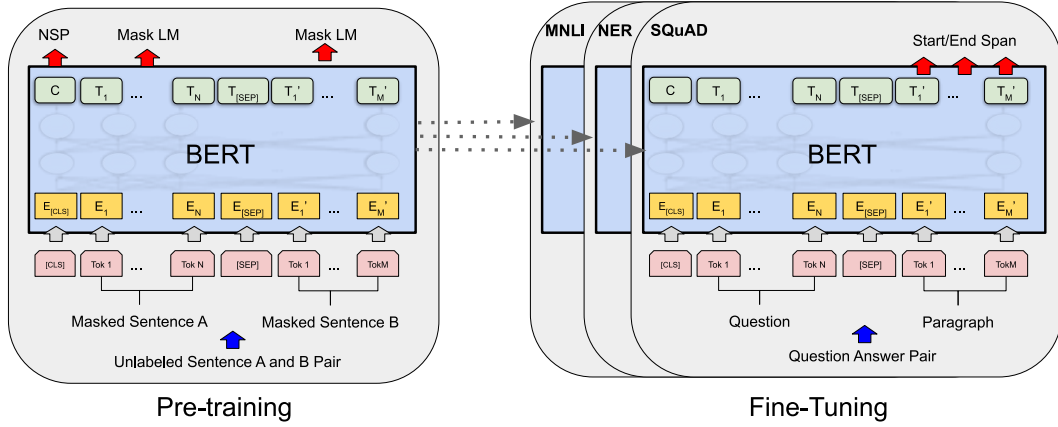


FIGURE 2.8: Pre-training and fine-tuning procedures for BERT.  
(Figure taken from Devlin et al. [2019] CC BY 4.0)

Devlin et al. [2019] tackle these problems using a different neural architecture. They propose *Bidirectional Encoder Representations from Transformers* (BERT)<sup>7</sup>. This model is based on a multi-layer bidirectional *Transformer* encoder [Vaswani et al., 2017], which allows to fully interconnect all tokens of a sequence using *self-attention* and parallelize the computations at the same time. In BERT, the input sequence is represented using word-piece embeddings [Wu et al., 2016] combined with positional embeddings [Gehring et al., 2017]. The latter is required because the Transformer architecture reduces complexity of dependencies by discarding information about the ordering of tokens in a sequence.

Training BERT for a task involves a two-step procedure (see Figure 2.8). The first step is *pre-training*, where the model is optimized for a self-supervised objective to gain basic language and domain understanding. Here, a masked language model is used to train bidirectional sentence representations. This is done by masking input tokens at random and then predict those masked tokens using softmax. Furthermore, a binarized next sentence prediction task is used to improve the understanding of relationships between sentences. The second step is *fine-tuning*, where task-specific inputs and outputs are used to continue training the model. Different to feature-based approaches such as ELMo, where fixed language representations are used as input to downstream tasks, fine-tuning involves the adaptation of all model parameters end-to-end. This second step is relatively inexpensive and requires less data than pre-training. Therefore, this procedure makes it possible to enhance task-specific models with pre-trained weights from domain-specific LMs, such as BioBERT [Lee et al., 2019].

In this thesis, we investigate the applicability of distributed representations, including neural LMs, to the information-seeking process. We focus on the desired model properties for Neural MR introduced in Section 1.2 and discuss the limitations of each model. Because a large amount of related work was published concurrently to this thesis, we subsequently include appearing work in each of the following chapters.

<sup>7</sup><https://github.com/google-research/bert>



## 2.3 Supervised Sequence Learning

One of our main objectives is to learn generalized Machine Reading models from unlabeled text. As we have seen in the previous section, distributional language models are often based on *self-supervision*, which is a process to apply *supervised learning* with training examples that can be obtained automatically using distributional statistics. Furthermore, we have identified and discussed the sequential properties of natural language. Subsequently, in this section we will discuss models and requirements for *supervised sequence learning* [Graves, 2012].

In general, a supervised learning task is based on a training set  $S$  and a test set  $S'$  that both contain input–target pairs  $(\mathbf{x}, \mathbf{y})$ . We assume that both  $S$  and  $S'$  are drawn from the same distribution. The goal of supervised machine learning is to minimize a specific error measure  $E$  defined on the test set by finding the maximum-likelihood parameters  $\Theta_{\text{ml}}$  for a model :

$$\Theta_{\text{ml}} = \arg \max_{\Theta} p(S | \Theta) = \arg \max_{\Theta} \prod_{(\mathbf{x}, \mathbf{y}) \in S} p(\mathbf{y} | \mathbf{x}, \Theta) \quad (2.18)$$

In neural networks, this objective is approximated by iteratively optimizing the parameters using a *loss function* on the training set, which is closely related to  $E$ . The standard procedure is to minimize the loss  $\mathcal{L}(S)$  defined as negative logarithm of the probability to predict the training examples:

$$\mathcal{L}(S, \Theta) = -\log \prod_{(\mathbf{x}, \mathbf{y}) \in S} p(\mathbf{y} | \mathbf{x}, \Theta) = - \sum_{(\mathbf{x}, \mathbf{y}) \in S} \log p(\mathbf{y} | \mathbf{x}, \Theta) \quad (2.19)$$

where the explicit dependence on  $\Theta$  is often left out in notation. The transfer of learning to predict  $S'$  from  $S$  is known as *generalization*.

When applying Machine Learning (ML) methods to sequential tasks, such as language modeling, we assume that the individual data points, such as words or sentences, are not independent [Graves, 2012]. Instead, both the input features  $\mathbf{x}_{1..T} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  and labels  $\mathbf{y}_{1..T} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$  constitute sequences with high correlation. Although linguistic input is not explicitly a time series, the individual inputs  $\mathbf{x}_t$  and labels  $\mathbf{y}_t$ ,  $t \in \{1, \dots, T\}$  are usually referred to as *time steps* at position  $t$ . Following this approach, we require a ML model to handle the alignment between individual time steps in the sequence. We will now discuss the most common sequence learning architectures with a focus on our Neural Machine Reading task.

### 2.3.1 Finite State Models

A simple approach to model sequence labeling is the *Hidden Markov Model* (HMM). This model assumes that a sequence of labels  $Y = (\mathbf{y}_1, \dots, \mathbf{y}_T)$  is generated by a *Markov process* with unobservable (hidden) states [Rabiner, 1989]. It further assumes that the observations  $X = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  are generated by another process which depends on  $Y$ . The goal is then to learn a

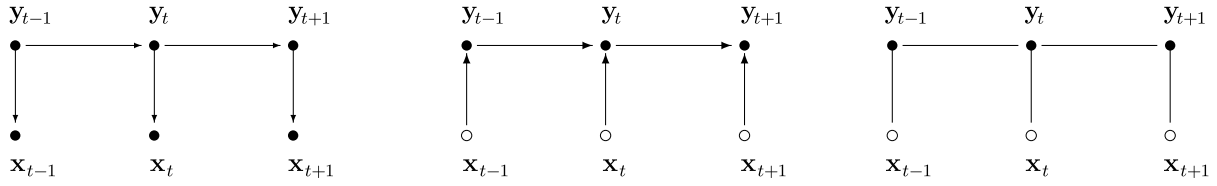


FIGURE 2.9: Dependency graph of simple HMMs (left), MEMMs (center) and CRF (right). Arrows show dependencies between states. States shown with open circles are not generated by the model. (Figure adapted from Lafferty et al. [2001])

generative model for  $X$ :

$$P(X) = \sum_Y P(X|Y)P(Y) \quad (2.20)$$

where the sum runs over all possible sequences of  $Y$ . This optimization problem can be solved by efficient dynamic programming algorithms such as forward-backward, Viterbi [Forney, 1973] or Baum-Welch [Baum et al., 1970]. However, these algorithms are especially not practical for feature-rich and long sequences. The most significant drawback of this model is that predictions for the next time step  $t + 1$  are solely based on the current time step  $t$  (see left side of Figure 2.9). Thus, the model ignores the previous history of the sequence entirely.

Therefore, *Maximum Entropy Markov Models* (MEMMs) extend this idea using a feature-based discriminative model. MEMMs describe the probability for reaching a label sequence  $Y$  using an initial state distribution  $P_0(Y)$  and a transition function. This function  $P_{Y'}(Y|X)$  describes the probability of reaching  $Y$  from a previous state  $Y'$  and the current observation sequence  $X$  (see center of Figure 2.9). It can be fitted per state using maximum entropy classifiers [McCallum et al., 2000].

MEMMs and other discriminative finite-state models share a *label bias problem*: Transition probabilities are calculated per state, and therefore they compete against each other, rather than against all transitions in the model. As a result, states with fewer outgoing transitions are preferred during optimization. *Conditional Random Fields* (CRF) addresses this problem by modeling the joint probability of the entire label sequence  $Y$  in a single exponential model [Lafferty et al., 2001]. In CRF,  $Y$  is represented as an undirected graph where vertices and edges are constructed using the Markov property. The joint distribution over the label sequence  $Y$  given  $X$  is expressed as a random field, which is optimized by maximizing the log-likelihood objective:

$$p(Y|X; \lambda) = \frac{\exp \sum_{t=1}^T \sum_j \lambda_j f_j(X, t, y_{t-1}, y_t)}{\sum_{y \in Y} \sum_{t=1}^T \sum_j \lambda_j f_j(X, t, y'_{t-1}, y'_t)} \quad (2.21)$$

where  $\lambda$  is a set of weights and  $f$  is a feature function that returns  $j$  feature values. The CRF model is much more expressive than HMM-like models, because it allows arbitrary dependencies on the observed sequence (see right side of Figure 2.9). Furthermore, in CRF, a state does not need to be specified completely by the features, therefore it can better exploit sparse training sequences [Lafferty et al., 2001].

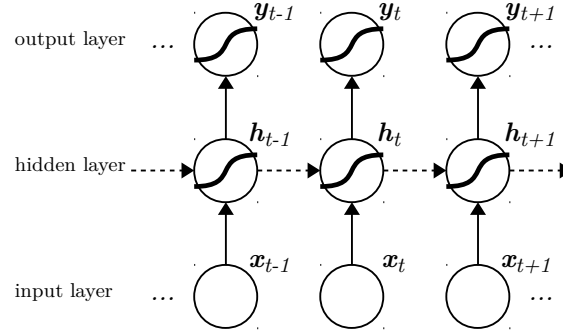


FIGURE 2.10: RNN architecture with input activations  $\mathbf{x}$ , hidden layer  $\mathbf{h}$  and output layer  $\mathbf{y}$ . Dotted lines depict recurrent connections.  
(Figure adapted from Graves [2012])

### 2.3.2 Recurrent Neural Networks

*Recurrent neural networks* (RNNs) are a class of neural architectures that handle sequential iterations using cyclical connections [Graves, 2012]. These recurrent connections allow the internal state of a network to act as ‘memory’ for previous inputs. A typical RNN architecture uses nodes that receive input from the current data point  $\mathbf{x}_t$  and from hidden node outputs  $\mathbf{h}_{t-1}$  from the previous time step. Here, nodes use nonlinear sigmoid activations  $\sigma$ . The network’s output  $\mathbf{y}_t$  is calculated from the hidden node values  $\mathbf{h}_t$  at time step  $t$  [Lipton and Berkowitz, 2015]:

$$\begin{aligned}\mathbf{h}_t &= \sigma(\mathbf{W}_{hx}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \\ \mathbf{y}_t &= \text{softmax}(\mathbf{W}_{yh}\mathbf{h}_t + \mathbf{b}_y)\end{aligned}\tag{2.22}$$

where  $\mathbf{W}_{hx}$ ,  $\mathbf{W}_{hh}$  and  $\mathbf{W}_{yh}$  are weight matrices for input, recurrent and output connections respectively, and  $\mathbf{b}_h$ ,  $\mathbf{b}_y$  are bias parameters. This architecture is visualized in Figure 2.10 with unfolded time steps. In practice, this network can be trained over many time steps using the *backpropagation through time* (BPTT) algorithm [Werbos, 1990].

### 2.3.3 Long Short-Term Memory

Training RNNs over long sequences is challenging, because gradients may vanish or explode along the large number of recurrent connections. This makes it especially hard to capture long-range dependencies in the data. Hochreiter and Schmidhuber [1997] introduce the *Long Short-Term Memory* (LSTM) model to overcome this problem. In contrast to RNNs, the hidden node is replaced by a *memory cell* (see Figure 2.11). Each cell contains five nodes, which are internally connected [Lipton and Berkowitz, 2015]:

- *Input node* ( $\mathbf{g}$ ): takes the activations from input layer  $\mathbf{x}_t$  and the previous time step  $\mathbf{h}_{t-1}$ , similar to a RNN. Typically, inputs are summed and run through a *tanh* activation  $\phi$ .

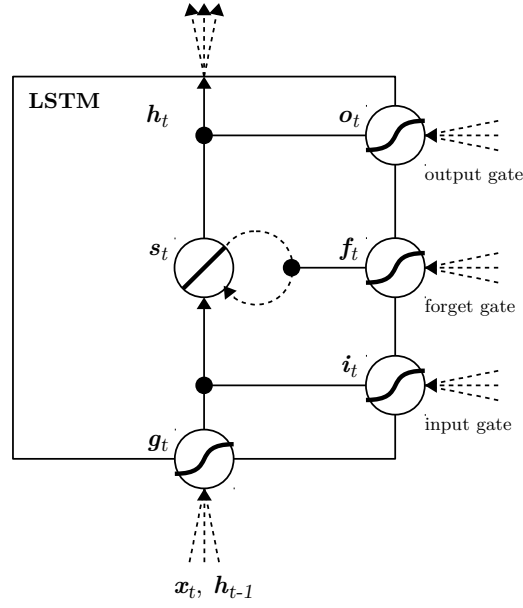


FIGURE 2.11: A single LSTM memory cell with input node  $g$ , internal state  $s$ , input gate  $i$ , forget gate  $f$  and output gate  $o$ . Dotted lines depict recurrent connections. (Figure adapted from Lipton and Berkowitz [2015])

- *Internal state (s)*: acts like a memory using a self-connected recurrent edge with linear activation. This ensures that the gradient can pass many time steps without vanishing or exploding.
- *Input gate (i)*: controls the flow from input node to internal state by multiplication with a sigmoid  $\sigma$ . If the gate value is zero, activations from other nodes are cut off. If the value is one, activations are passed through.
- *Forget gate (f)*: enables the cell to delete its internal memory [Gers et al., 2000]. If the gate value is zero, the internal state is flushed. If the value is one, the internal state is kept.
- *Output gate (o)*: controls the flow from internal state to the output  $h_t$ .

The computation algorithm for one LSTM layer is as follows [Lipton and Berkowitz, 2015]:

$$\begin{aligned}
 g_t &= \phi(\mathbf{W}_{gx}\mathbf{x}_t + \mathbf{W}_{gh}\mathbf{h}_{t-1} + \mathbf{b}_g) \\
 i_t &= \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i) \\
 f_t &= \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f) \\
 o_t &= \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o) \\
 s_t &= \phi(g_t \odot i_t + s_{t-1} \odot f_t) \\
 h_t &= s_t \odot o_t
 \end{aligned} \tag{2.23}$$

where  $\mathbf{W}$ ,  $\mathbf{b}$  are weight matrices and bias terms, and  $\odot$  denotes element-wise multiplication.

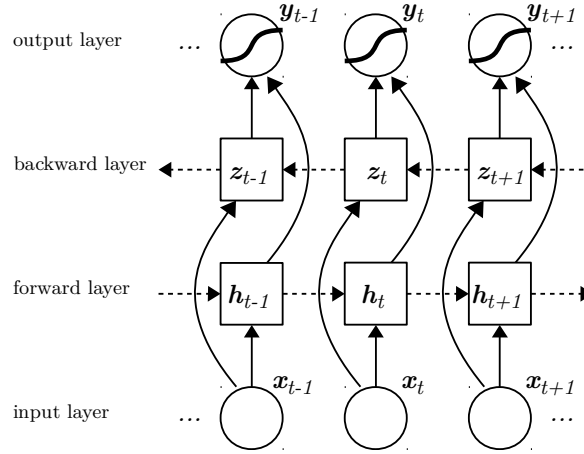


FIGURE 2.12: Bidirectional LSTM architecture with input activations  $\mathbf{x}$ , forward layer  $\mathbf{h}$ , backward layer  $\mathbf{z}$  and output layer  $\mathbf{y}$ .  
(Figure adapted from Graves [2012])

For many tasks it is beneficial to extend this unidirectional LSTM so that it can handle information from left and right context of the current time step. Therefore, the *bidirectional LSTM* (BLSTM) uses a second independent backward layer  $\mathbf{z}$  (see Figure 2.12). The backward layer is defined similarly to above, but with recurrent connections  $\mathbf{z}_{t+1}$  instead of  $\mathbf{h}_{t-1}$ . Both layers are combined to train a common objective  $\mathbf{y}$  [Lipton and Berkowitz, 2015]:

$$\mathbf{y}_t = \text{softmax}(\mathbf{W}_{yh}\mathbf{h}_t + \mathbf{W}_{yz}\mathbf{z}_t + \mathbf{b}_y) \quad (2.24)$$

where  $\mathbf{W}_{yh}$ ,  $\mathbf{W}_{yz}$  are weight matrices  $\mathbf{b}_y$  is a bias term. The BLSTM can be trained the same way as an RNN using BPTT, or—in case of very long or running sequences—with *truncated backpropagation through time* (TBPTT) [Williams and Zipser, 1989].

### 2.3.4 Transformer

One inherent problem of recurrent models, such as RNN and LSTM, is that they cannot efficiently be parallelized. Recurrent models require  $O(n)$  sequential operations to compute an example of  $n$  time steps. Handling long-range dependencies is especially expensive in recurrent models, because the maximum path length to compute between two examples is  $O(n)$  as well. Vaswani et al. [2017] introduce the *Transformer* to overcome this problem. Their model is entirely based on attention to draw global dependencies between input and output. The Transformer can efficiently parallelize computations by keeping a maximum path length of  $O(1)$  and therefore reducing sequential operations to  $O(1)$ .

The Transformer's architecture is shown in Figure 2.13. It consists of  $N$  layers of *encoder* and *decoder* stacks. The input is a sequence of token embeddings concatenated with positional encodings, which provide ordering information of the tokens. Similar to sequence-to-sequence models, this architecture allows every position in the decoder to attend over all positions in the input sequence. An encoder is composed of a self-attention mechanism on the input, and

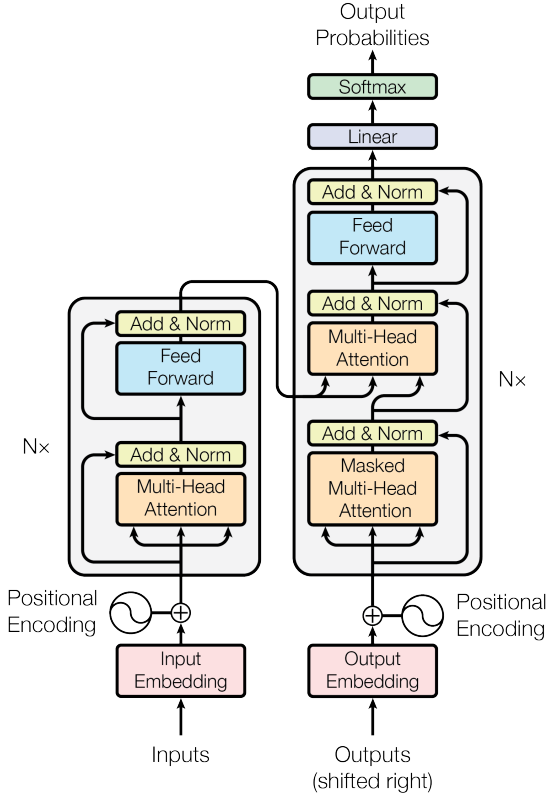


FIGURE 2.13: Architecture of the Transformer model with encoder (left) and decoder stacks (right). (Figure taken from Vaswani et al. [2017])

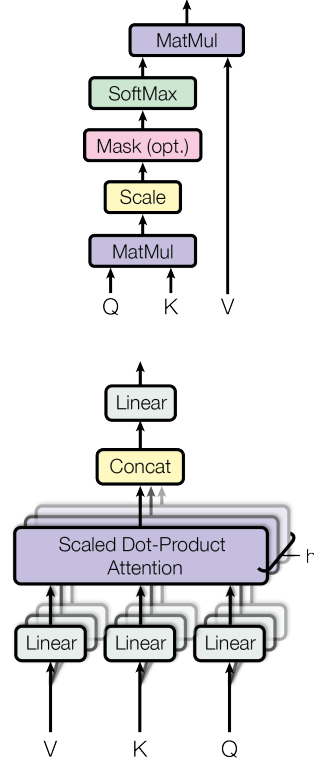


FIGURE 2.14: Architecture of scaled dot-product attention (top) and multi-head attention (bottom). (Figures taken from Vaswani et al. [2017])

a feed-forward network, with residual connections and layer normalization after each step. A decoder is composed of masked attention on the output, self-attention over the decoder state and the encoder output, and a feed-forward network, again with residual connections and normalization after each step. The masking is necessary to ensure that predictions depend only on the known outputs from previous time steps. The decoder produces a logit vector containing the softmax probabilities for the task output. Typically,  $N = 6$  layers are used with an embeddings dimension of 512.

The Transformer uses a *scaled dot-product attention* mechanism, which calculates alignment values  $V$  between input keys  $K$  of dimension  $d_k$  and output queries  $Q$  [Vaswani et al., 2017]:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.25)$$

In *multi-head attention*, this function is performed  $h$  times in parallel with different learned linear projections of queries, keys and values. This allows the model to jointly attend to different positions from the representation subspaces. The attention mechanisms are shown in Figure 2.14. Typically,  $h = 8$  attention heads with 64 dimensions are used.

A Transformer model with dimension  $d$  has a total computational complexity per layer of  $O(n^2 \cdot d)$ , in contrast to RNN, which has a complexity of  $O(n \cdot d^2)$  [Vaswani et al., 2017]. Subsequently, Transformer models not only provide better parallelization, but are also better scalable in memory capacity, especially for short sequences where  $n < d$ . However, a large drawback of Transformers is the memory requirement for long sequences, such as entire documents with thousands of tokens.

## 2.4 Discussion

In this chapter, we have discussed traditional approaches for Information Extraction, which are often based on discrete syntactic processing. We have shown that these models improve from pre-trained distributed language representations, but—most prominently in the case of Entity Linking—remain inside the symbolic paradigm and generate discrete outcomes. The vision of Neural Machine Reading, however, is to design the information-seeking process as a continuous end-to-end task. In this scenario, intermediate representations are distributed and allow semantic nuances to propagate through multiple layers in the application.

Sequence labeling models and Transformers, on the other hand, are currently too focused on the distributional hypothesis. Especially in specific domains such as clinical medicine, a large amount of knowledge is modeled using hierarchical structure, for example in the Unified Medical Language System (UMLS). A domain-specific Neural Machine Reading model needs to utilize this structural knowledge in the same way as neural Language Models utilize distributional information from raw text. Furthermore, in long documents the distributional hypothesis may fail, because long-range and multi-document dependencies can not effectively be resolved by these models.

In this thesis, we therefore keep on following the traditional ideas of word-sense disambiguation, coreference resolution, Entity Linking and Topic Modeling. However, we do not primarily regard them as linguistic tasks on their own. Instead, we aim to embed these tasks into the larger picture of the information-seeking process.





## Chapter 3

# A Robust Model for Efficient Entity Linking

In Information Extraction, named entities are the smallest units of information [Sarawagi, 2008]. Therefore, we expect from a Machine Reading model to be able to detect and represent entities in a way so they can be linked to an existing knowledge base. In this chapter, we approach RQ 1: *What are general solutions to identify named entities in domain-specific text?*

We present TASTY, a robust end-to-end model for Named Entity Recognition (NER) and Linking (NEL)<sup>1</sup>. As we aim to create a MR model for domain-specific text resources, we put a special focus on vertical corpora, such as Reuters news, Frankfurter Rundschau, Medline biomedical abstracts and car discussion forums. These corpora contain a large amount of domain-specific entities with potentially idiosyncratic names and context from different languages. We approach the NER task with contextual sequence learning and combine subword-level token representations combined with distributed word embeddings. Our NEL approach utilizes cosine similarity between model’s predictions and entity embeddings. We design TASTY so that it is efficiently trainable with only few hundred labeled sentences and target to reach state-of-the-art results at the same time.

This chapter is structured as follows: In Section 3.1, we introduce the tasks of NER and NEL, their design challenges and common errors. In Section 3.2, we describe our TASTY Entity Linking model, which uses robust word encoding (Section 3.2.1), a sequence learning based approach for recognition (Section 3.2.2) and a vector space approach for disambiguation (Section 3.2.3). In Section 3.3, we evaluate both tasks using publicly available datasets from different domains in English and German. We show that the effectiveness of our TASTY model is on par with state-of-the-art and highlight its efficiency in scenarios with limited available training data. In Section 3.4, we analyze errors produced by our model. We summarize this chapter in Section 3.5 with a review of the research questions posed at the beginning and we highlight the building blocks that will be important during the following chapters of this thesis.

---

<sup>1</sup>The main parts of this chapter were published by S. Arnold, F. A. Gers, T. Kiliyas, and A. Löser [2016b]. “Robust Named Entity Recognition in Idiosyncratic Domains”. In: *arXiv:1608.06757 [cs.CL]*. The chapter contains additional results published by S. Arnold, R. Dziuba, and A. Löser [2016a]. “TASTY: Interactive Entity Linking As-You-Type”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 111–115.

### 3.1 Introduction

Information Extraction tasks have become very important not only in the Web, but also for in-house enterprise settings. One of the crucial steps towards understanding natural language is Named Entity Linking (NEL), which aims to extract mentions of entity names in text and link them to a knowledge base. NEL is necessary for many higher-level tasks such as Relation Extraction, Knowledge Base Population, Question Answering and Information Retrieval. In these scenarios recall is critical, because candidates that are not generated by an NEL system can not be recovered later [Hachey et al., 2013].

We contribute TASTY, a general annotator for robust Named Entity Recognition and Linking that can be trained for vertical domains with low human labeling effort. TASTY does not depend on domain-specific rules, dictionaries, fine-tuning, syntactic annotation or external knowledge. Instead, our approach is built as an end-to-end model which is based on low-level character features of text. We train our model for news and biomedical domains with raw text data and few hundred labels. With this model, we aim to achieve equal performance compared to state-of-the-art NER annotators for common tasks, such as CoNLL2003, KORE50, ACE2004 and MSNBC. We further apply the highly domain-specific biomedical GENIA corpus and a car discussion forum dataset to show how our approach adapts to various idiosyncratic domains. In particular, we observe that letter-trigram word encoding with surface form features efficiently compensates typing and capitalization errors and Bidirectional Long Short-Term Memory (BLSTM) networks capture useful distributional context required for effective NER. With a combination of these techniques, we achieve better context representation than word2vec models trained with significantly larger corpora.

#### 3.1.1 Task Definition

The Entity Linking task aims to establish links between a dataset of text documents  $\mathcal{D}$  and a structured knowledge base containing named entities  $e_j \in \mathcal{K}$  [Ji and Grishman, 2011]. This problem can be divided into two stages [Hachey et al., 2013]. In the mention recognition (NER) stage, the goal is to detect a set of entity mentions  $m_i \in \mathcal{M}_d$  for each document  $d \in \mathcal{D}$ , so that each mention holds a span of tokens that refers to an entity or concept. In the disambiguation (NED) stage, the goal is to create the entity links  $\mathcal{E}_d$  by assigning to each of the mentions an entry of the knowledge base:

$$\mathcal{E}_d = \{\langle m, e \rangle \mid m \in \mathcal{M}_d \wedge \exists e \in \mathcal{K} : m \text{ refers to } e\} \quad (3.1)$$

This further implies that mentions which do not refer to an entry in  $\mathcal{K}$  (often called *NIL*) are not contained in  $\mathcal{E}_d$ . We will focus on these two stages in sections 3.2.2 and 3.2.3.

### 3.1.2 Challenges for Entity Linking

Pink et al. [2014] show that NER components can reduce the search space for slot filling tasks by 99.8% with a recall loss of 15%. However, large effort is required to adapt most annotators to specialized domains, such as biomedical documents. When focusing on recall for these domains, we face three major problems. First, the language used in the documents is often idiosyncratic and cannot be effectively identified by standard NLP tools [Prokofyev et al., 2014]. Second, training these domains is difficult: data is sparse, data may contain a large number of non-linkable entity mentions (NILs) and large labeled gold standards are hardly available. Third, applications vary greatly and we cannot standardize annotation guidelines to meet all of their requirements [Ling et al., 2015]. For example, NER on news texts might focus on proper named entity annotation (e.g. people, companies and locations), whereas phrase recognition on medical text might include the annotation of common concepts (e.g. medical terms and treatments). We therefore focus on building a *generalized* system with high recall, which can be efficiently trained with only few labeled examples.

### 3.1.3 Common Error Analysis

Ling et al. [2015] point out common errors of NER systems, which yield non-recognized mentions (false negatives), invalid detections (false positives), wrong boundaries (e.g. multi-word mentions, missing determiners) and annotation errors from human labelers (e.g., correct answers are not marked as correct, unclear annotation guidelines). Consider the following example taken from the biomedical GENIA corpus [Kim et al., 2003], with underlined named entity mentions:

**Example 3.1.** Engagement of the Lewis X antigen (CD15) results in monocyte activation. Nuclear extracts of anti-CD15 cross-linked cells demonstrated enhanced levels of the transcriptional factor activator protein-1, minimally changed nuclear factor-kappa B, and did not affect SV40 promoter specific protein-1.

We observe that common errors are caused by a manifold number of frequent factors:

- non-verbatim mentions (e.g. misspellings, alternate writings: monocytes, Lewis-X)
- part-of-speech (POS) tagging errors (e.g. unidentified NP tags: [monocyte]<sub>JJ</sub>)
- wrong capitalization (e.g. uppercase headlines, lowercase proper names)
- unseen or novel words (e.g. idiosyncratic language: anti-CD15)
- irregular word context (e.g. collapsed lists, semi-structured data, invalid segmentation)

We therefore focus on building a *robust* system, which does not rely on linguistic preprocessing, but instead is trained with in-domain end-to-end data and character-based representations.

## 3.2 Entity Linking Model

In this section we describe the three stages of our TASTY Entity Linking model: robust word encoding, Named Entity Recognition and Named Entity Disambiguation.

### 3.2.1 Robust Word Encoding

We have shown that the most common errors for recall loss are misspellings, POS errors, capitalization, unseen words and irregular context. Therefore we carefully design the input representations to our model using three complementary feature spaces: distributed word embeddings, letter-trigram representations and surface form features.

**Word embeddings.** Sahlgren [2008] proposes local word context features for resolving *paradigmatic relations* (e.g. cyclosporin A-treated cells / HU treated cells). We apply to this problem the efficient technique of Mikolov et al. [2013a]. Their approach utilizes the continuous Skip-gram model to classify a word based on the distribution of other words in the same sentence. Our implementation is based on word2vec and represents words in dense vector space.

$$\mathbf{x}_{\text{emb}}(w) = \text{word2vec}(w) \quad (3.2)$$

**Letter-trigram hashing.** Dictionary-based word vectorization methods suffer from sparse training sets, especially in the case of non-verbatim mentions, rare words, spelling and capitalization errors. For example, word2vec generalizes insufficiently for rare words in idiosyncratic domains or for misspelled words, since for these words no vector representation is learned at training time. In the biomedical GENIA data set, we notice 27% unseen words (out-of-vocabulary) in the pretrained model. As training data generation is expensive, we investigate a general approach for the generation of word vectors. We use letter-trigram word hashing as introduced by Huang et al. [2013]. This technique splits a word into discriminative three-letter ‘syllables’ with boundary markers and generates an n-hot vector using this bag, e.g. “cell”  $\rightarrow \{\#ce, cel, ell, ll\# \}$  (see Figure 3.1). With this partitioning, the vector is robust against misspellings and out-of-vocabulary words and has the advantage to group similar morphologic words in similar vector spaces:

$$\mathbf{x}_{\text{tri}}(w) = \sum_{t \in \text{trigram}(w)} \mathbf{e}_{\text{idx}(t)} \quad (3.3)$$

where  $\text{idx}(t)$  is a function that returns the index of a trigram and  $\mathbf{e}_i$  is the  $i$ -th unit vector.

**Surface form features.** Words appear in various surface forms, e.g. capitalized at the beginning of sentences, uppercase in headlines, lowercase in social media text or other erroneous capitalizations. However, the most important features for shallow learners are word shape properties, such as length, initial capitalization, all-word uppercase, in-word capitalization

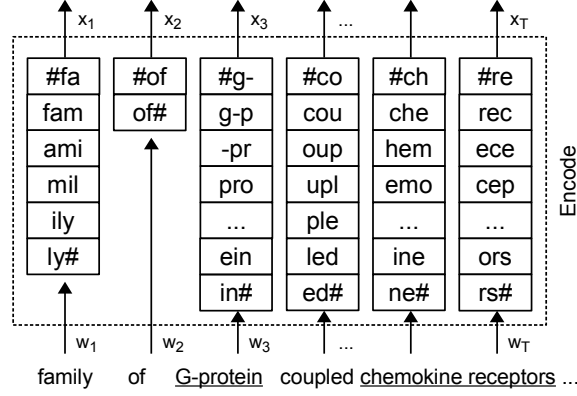


FIGURE 3.1: Architecture of robust letter-trigram word encoding. The character stream “family of G-protein coupled chemokine receptors” is tokenized into words  $w_{1...T}$  and encoded into word vectors  $x_{1...T}$  using letter-trigram hashing.

and use of numbers or punctuation [Ling and Weld, 2012]. When using mixed-case dictionaries, these features are implicitly included in the encoding. We achieve stronger generalization and faster learning by encoding words as lowercase and isolating the surface information as 15 boolean values in the feature vector that indicate uppercase (UC), lowercase (LC), numerics (NUM), punctuation (PUC), document (DOC) and sentence (SNT) features:

$$\mathbf{x}_{sf}(w) = \begin{bmatrix} \text{startUC}(w), \text{allUC}(w), \text{startLC}(w), \text{allLC}(w), \text{mixedCase}(w), \\ \text{startNUM}(w), \text{someNUM}(w), \text{allNUM}(w), \text{endNUM}(w), \\ \text{startPUC}(w), \text{endPUC}(w), \\ \text{startSNT}(w), \text{startDOC}(w), \text{endSNT}(w), \text{endDOC}(w) \end{bmatrix}^T \quad (3.4)$$

### 3.2.2 Named Entity Recognition

We model mention recognition as sequential word labeling problem. We express each sentence in a document as a sequence of words:  $s = (w_1, \dots, w_T)$ . We define a mention as the longest possible span of adjacent tokens that refer to an entity or relevant concept of a real-world object, such as CD28 surface receptor. We further assume that mentions are non-recursive and non-overlapping. Figure 3.2 illustrates an example for context-sensitive transformation of a word sequence  $s = (w_1, \dots, w_T)$  into word labels  $y = (y_1, \dots, y_T)$ . We use the BIOES tagging scheme [Ratinov and Roth, 2009] to encode boundaries of a mention span. We assign labels  $\{B, I, O, E, S\}$  to each token to mark begin (B), inside (I), outside (O), end (E) and single-word (S) mentions, reading from left to right. Our objective is now to predict the most likely label  $\hat{y}_t$  for a word  $w_t$  regarding its context:

$$\hat{y}_t = \arg \max_{l \in \{B, I, O, E, S\}} P(y_t = l \mid w_1, \dots, w_{t-1}, w_t, w_{t+1}, \dots, w_T) \quad (3.5)$$

We utilize recurrent neural networks to approximate a solution for this objective.

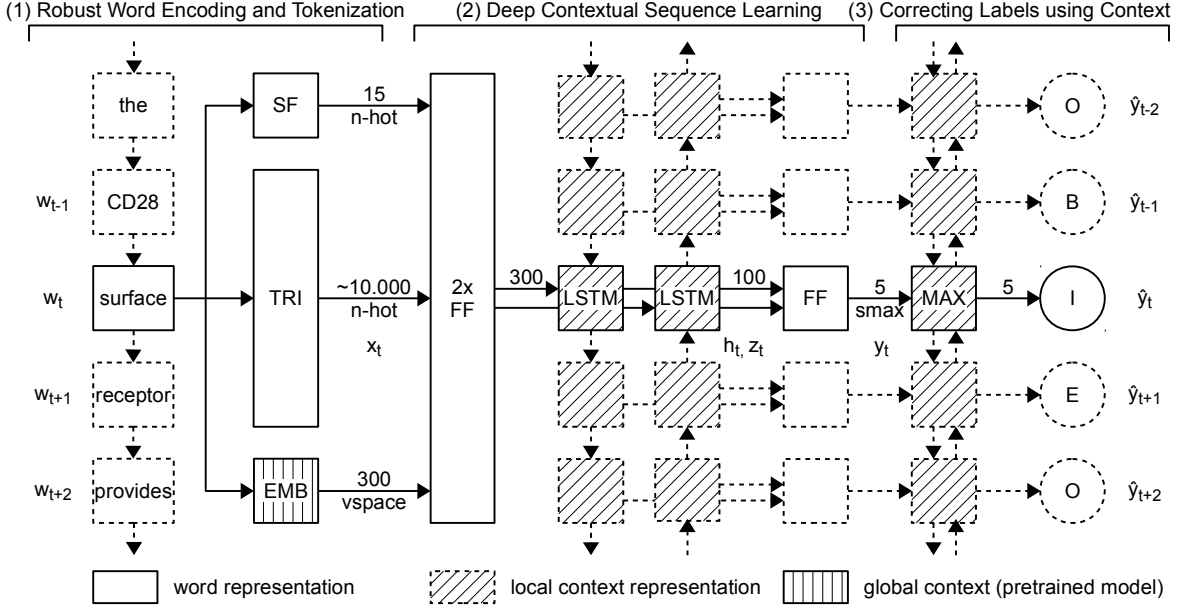


FIGURE 3.2: Architecture of the BLSTM network used for named entity recognition. (1) The character stream “the CD28 surface receptor provides” is tokenized into words and converted into word feature vectors using surface forms (SF), letter-trigram hashing (TRI) and word embeddings (EMB). (2) We use a deep contextual sequence learner with stacked feed-forward (FF) and recurrent (LSTM) layers for bidirectional context representation. (3) We correct BIOES word labels using local context and decode them into mention annotations.

**Sequence learning.** To efficiently recognize mentions in text, long-range context-sensitive information is indispensable. Especially in the idiosyncratic domain, we expect noisy input data with high variance. Thus, we require strong generalization not only for the syntactic representation of language, but also for the latent semantic dependencies between the words in a document. We approach this problem by applying the computational model of recurrent neural networks, in particular Long Short-Term Memory networks (LSTMs) [Hochreiter and Schmidhuber, 1997] with forget gates [Gers et al., 2000] to the problem of sequence learning. Like deep neural feed-forward (FF) networks, LSTMs are able to learn complex parameters using gradient descent, but include additional recurrent connections between cells to influence weight updates over adjacent time steps. With their ability to memorize and forget over time, LSTMs have proven to generalize context-sensitive sequential data well [Graves, 2012; Lipton and Berkowitz, 2015].

**Bidirectional Long Short-Term Memory.** We use a combination of all three word feature approaches as input for a network of five stacked layers (see Figure 3.2). The word-wise sequential input vectors  $\mathbf{x}_{1..T}$  are calculated using concatenation ( $\oplus$ ):

$$\mathbf{x}(w) = \mathbf{x}_{\text{emb}}(w) \oplus \mathbf{x}_{\text{tri}}(w) \oplus \mathbf{x}_{\text{sf}}(w) \quad (3.6)$$

We squash the input vector using two fully connected FF layers of size 300. We utilize the efficient rectified linear unit (ReLU) activation, which is more robust against the vanishing gradient problem [Nair and Hinton, 2010]. The core sequence learner is composed of two LSTM layers with 100 cells each, one connected to read the sentence from left to right, the other in reverse direction. The bidirectional layers with forward state  $\mathbf{h}_t$  and backward state  $\mathbf{z}_t$  are combined into a final FF layer with 5-class softmax activation. The output of this layer  $\mathbf{y}_t$  is a probability distribution over BIOES labels per time step  $t$ :

$$\begin{aligned}\mathbf{h}_t &= \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t) \\ \mathbf{z}_t &= \text{LSTM}(\mathbf{h}_{t+1}, \mathbf{x}_t) \\ \mathbf{y}_t &= \text{softmax}(\mathbf{W}_{yh}\mathbf{h}_t + \mathbf{W}_{yz}\mathbf{z}_t + \mathbf{b}_y)\end{aligned}\tag{3.7}$$

where  $\mathbf{W}_{yh}$ ,  $\mathbf{W}_{yz}$  are weight matrices and  $\mathbf{b}_y$  is a bias term. We iterate over labeled sentences as training examples in mini-batches to update weights and bias parameters using backpropagation through time [Werbos, 1990]. The network is then used to predict label probabilities  $\mathbf{y}_{1...T}$  for unseen word sequences  $w_{1...T}$ . We implement the network using the DL4J<sup>2</sup> framework. We use stochastic gradient descent with RMSProp [Tieleman and Hinton, 2012] and a learning rate of 0.0025 with L2 regularization. Using the bidirectional LSTM, we achieve deeper contextual understanding, e.g. over the boundaries of multi-word annotations and at the beginning of sentences.

**Entity mention decoding.** To apply the sequential sequence learner to unseen text, we do a forward pass through the network to predict label probabilities  $y = (y_1, \dots, y_T)$  for each sentence. The output of the BLSTM softmax is a local per-word probability vector  $y_t$  for BIOES labels. However, this prediction cannot guarantee a correct order of labels according to the BIOES scheme, e.g.m E is only valid after B or I. We generate a corrected label sequence  $\hat{y}$  for each sentence  $\tilde{y} = (\text{O}, y_1, \dots, y_T, \text{O})$  by maximizing the combined probability of all possible valid sequences:

$$\hat{y} = \arg \max_y \prod_{t=2}^{N+1} \psi(\tilde{y}_{t-1}, \tilde{y}_t)\tag{3.8}$$

where  $\psi(y_i, y_j) = 1$ , if  $y_j$  is a valid label after  $y_i$ ; 0 otherwise. This step is usually implemented as CRF classifier [McCallum and Li, 2003]. However, we rely on the more generalized LSTM sequence learner and use this simple approach for correction. Since  $\psi$  only depends on two adjacent tokens, we implement the optimization of Equation 3.8 using dynamic programming. Finally, we decode the corrected sequence  $\hat{y}$  into a set of sentence-level mention annotations  $\mathcal{M}_s$  and add the instances to the document mentions  $\mathcal{M}_d$ .

<sup>2</sup><http://deeplearning4j.org> version 0.4.0

### 3.2.3 Named Entity Disambiguation

The next step is to disambiguate the set of mentions to the knowledge base  $\mathcal{K}$ . We break down this problem into two steps. In the first step, *candidate retrieval*, we retrieve a set of candidates  $\mathcal{C}_m$  for each mention  $m \in \mathcal{M}_d$ . In the second step, *disambiguation* we aim to find the most likely entity assignment  $\langle m, e \rangle$  by ranking all candidates and picking the first-ranked candidate.

**Knowledge Base.** The knowledge base  $\mathcal{K}$  is constructed as a set of entities, which we represent as a tuple:

$$\mathcal{K} = \{e \mid e = \langle \text{id}, \mathcal{N}, \epsilon, \sigma \rangle\} \quad (3.9)$$

where  $\text{id}$  is a unique entity identifier,  $\mathcal{N}$  is a list of names and abbreviations for the entity,  $\epsilon$  is an entity name embedding and  $\sigma$  a context embedding. We populate this information by parsing page names, page titles and anchor links from a recent English Wikipedia dump<sup>3</sup>. We store this information in a Lucene<sup>4</sup> to allow lookup and search of entity names with low latency.

**Entity embeddings.** For each entity  $e$  in the knowledge base, we create two distributed entity embeddings  $\epsilon_e$  and  $\sigma_e$ . The purpose of these vectors is to encode contextual information for disambiguation. The *entity name embedding* encodes all known names for an entity using the trigram representation:

$$\epsilon_e = \sum_{w \in \mathcal{N}_e} \sum_{t \in \text{trigram}(w)} \mathbf{e}_{\text{id}x(t)} \quad (3.10)$$

The *entity context embedding* encodes typical entity context  $c_e$  as distributed vector representation. We utilize the Paragraph Vectors model [Le and Mikolov, 2014]:

$$\sigma_e = \text{ParVec}(c_e) \quad (3.11)$$

We train this model using Wikipedia articles and generate the entity context embeddings using the abstract of each entity.

**Candidate retrieval.** In the first step, we generate entity candidates by matching the mention string to the set of names contained in the alias list:

$$\mathcal{C}_m = \{e \mid e \in \mathcal{K} \wedge m \in \mathcal{N}_e\} \quad (3.12)$$

We use the implementation of BM25 in Lucene with additional fuzzy matching parameters to allow higher recall.

<sup>3</sup><https://dumps.wikimedia.org/enwiki/>

<sup>4</sup><https://lucene.apache.org>



**Disambiguation.** In the second step, we aim to find the most likely entity assignment  $\langle m, e \rangle$  by ranking all candidates and picking the first-ranked candidate:

$$\mathcal{E}_d = \{ \langle m, e \rangle \mid m \in \mathcal{M}_d \wedge e = \arg \max_{c \in \mathcal{C}_m} \text{sim}(c, m, d) \} \quad (3.13)$$

The similarity function  $\text{sim}(c, m, d)$  uses cosine similarity between the pre-calculated entity embeddings of each candidate  $\epsilon_c, \sigma_c$ , the mention name  $\epsilon_m$  and document context  $\sigma_d$ :

$$\text{sim}(c, m, d) = \text{cosine}(\epsilon_c \oplus \sigma_c, \epsilon_m \oplus \sigma_d) \quad (3.14)$$

While picking the first-ranked candidate could be replaced by collective re-ranking over a document, we observe that the integration of document context in the similarity function provides enough contextual information to achieve coherent disambiguation.

### 3.3 Evaluation

We evaluate two configurations of our TASTY model on eleven gold standard evaluation data sets. We perform Named Entity Recognition and Named Entity Disambiguation tasks. We show that the combination of letter-trigram word hashing and word embeddings with bidirectional LSTM yields the best results and outperforms sequence learners based on dictionaries or word2vec. To highlight the generalization of our model to specialized domains, we run tests on common-typed English data sets as well as on German and English datasets from the biomedical and car industry domains. We compare our system on these data sets with state-of-the-art entity linkers and annotators from NLP pipelines.

#### 3.3.1 Evaluation Set-up

We train two models with identical parameterization, each with labeled sentences from the corresponding training data set. The first model, TASTY, is solely based on letter-trigram features from the training data. The second model, TASTY+emb, utilizes additional word embeddings to include distributional context information from pre-training a larger dataset<sup>5</sup>. For preprocessing (sentence splitting and word tokenization), we use Stanford CoreNLP 3.6.0 [Manning et al., 2014]. The prediction model was trained using Deeplearning4j with nd4j-x86 backend. Training TASTY on a 4-core Intel i7 CPU at 2.8GHz takes approximately 50 minutes.

**Evaluation datasets.** We use a variety of standard and domain-specific data sets for training and evaluation. *CoNLL2003* [Kim et al., 2004] is a standard NER dataset based on the Reuters RCV-1 news corpus. It covers named entities of type person, location, organization and miscellaneous in English and German languages. *KORE50* [Hoffart et al., 2012] use a similar annotation scheme on the same corpus. The *TIGER* treebank [Brants et al., 2002] is based on German

<sup>5</sup>we utilize the pretrained GoogleNews-vectors-negative300 embeddings model

Dataset	CoNLL03-en			KORE50			MSNBC			ACE2004		
domain task	English news named entities			English news named entities			English news wikification			English news concepts		
Model	Prec	Rec	$F_1$	Prec	Rec	$F_1$	Prec	Rec	$F_1$	Prec	Rec	$F_1$
Babelfy	44.2	62.7	51.8	69.2	68.8	69.0	36.9	66.2	47.4	8.2	48.0	14.1
DBpedia Spotlight	66.6	58.6	62.4	–	–	–	48.6	45.2	46.8	11.1	64.4	18.9
Entityclassifier	68.2	69.8	69.0	87.6	88.2	87.9	61.7	76.0	68.1	11.5	77.8	20.0
Stanford NER	<b>96.4</b>	73.6	83.5	91.4	73.6	81.5	<b>87.9</b>	77.6	<b>82.4</b>	17.1	85.6	<b>28.6</b>
LingPipe	69.0	50.3	58.2	78.0	71.5	74.6	53.1	57.0	55.0	12.7	69.6	21.5
FOX	94.7	71.3	81.3	90.4	71.5	79.8	3.2	2.7	2.9	–	–	–
NERD-ML	51.7	62.1	56.4	67.5	79.2	72.8	59.8	47.7	53.1	<b>19.6</b>	34.3	25.0
<b>TASTY</b>	87.9	90.1	89.0	<b>96.6</b>	<b>92.2</b>	<b>94.3</b>	68.5	<b>83.1</b>	75.2	11.7	<b>86.0</b>	20.5
<b>TASTY+emb</b>	90.3	<b>92.0</b>	<b>91.1</b>	95.3	<b>92.2</b>	93.7	72.4	82.7	77.2	–	–	–

TABLE 3.1: Experimental results for English Named Entity Recognition for four standard data sets compared to seven state-of-the-art annotators. The table shows micro-averaged Precision, Recall,  $F_1$  scores for exact annotation span match. All annotators use the same domain- and task-independent model across all experiments.

Dataset	CoNLL03-de			TIGER			GENIA			CAR-Model			CAR-Part		
domain task	German news named entities			German news noun phrases			English biomedical biomedical terms			German web car models			German web car parts		
Model	Prec	Rec	$F_1$	Prec	Rec	$F_1$	Prec	Rec	$F_1$	Prec	Rec	$F_1$	Prec	Rec	$F_1$
TagMe	–	–	–	–	–	–	–	–	–	2.2	29.8	4.0	1.3	21.4	2.5
Stanford NER	<b>89.4</b>	63.9	74.5	68.9	31.7	43.4	31.7	7.6	12.3	–	–	–	–	–	–
LingPipe	–	–	–	–	–	–	71.8	68.3	70.0	–	–	–	–	–	–
<b>TASTY</b>	88.1	87.6	87.8	81.6	71.3	76.1	75.7	<b>80.3</b>	77.9	74.6	72.8	73.7	69.6	66.9	68.2
<b>TASTY+emb</b>	87.3	<b>88.6</b>	<b>87.9</b>	<b>82.7</b>	<b>83.9</b>	<b>83.3</b>	<b>77.5</b>	79.5	<b>78.5</b>	<b>82.5</b>	<b>79.9</b>	<b>81.2</b>	<b>79.3</b>	<b>70.8</b>	<b>74.8</b>

TABLE 3.2: Experimental results for domain-specific Named Entity Recognition on five data sets compared to standard entity annotators (micro-averaged Precision, Recall,  $F_1$  scores). The datasets vary in language, source domain and annotation task definition. Therefore, all TASTY annotators are specifically trained using labeled in-domain data.

news from Frankfurter Rundschau and contains annotated noun phrases. *MSNBC* [Cucerzan, 2007] contains NEL annotations for linking English news texts to Wikipedia articles. *ACE2004* [Mitchell et al., 2005] contains NEL annotations covering named entities and general concepts on English news text. *AQUAINT* [Milne and Witten, 2008] contains NEL annotations on long English news documents, covering only the first mention of an entity. The *DBpedia Spotlight* corpus [Mendes et al., 2011] contains NEL annotations on English news articles. The *GENIA* Corpus [Kim et al., 2003] contains biomedical abstracts from PubMed and covers biomedical entities such as proteins, genes and cells. The *CAR* corpus was created by crawling German discussion forums on cars and annotating car models and parts [Mehlitz, 2019].

Configuration	CoNLL03-en			GENIA		
	Prec	Rec	$F_1$	Prec	Rec	$F_1$
<b>TASTY+emb</b>	<b>90.3</b>	<b>92.0</b>	<b>91.1</b>	<b>77.5</b>	79.5	<b>78.5</b>
no LSTM	-21.3	-20.1	-20.6	-15.1	-14.3	-14.8
unidirectional LSTM	-5.1	-2.0	-3.5	-2.6	-3.8	-3.3
1-hot encoding	-14.2	-16.4	-15.3	-6.4	+0.4	-3.2
no EMB encoding	-2.4	-1.9	-2.1	-1.8	<b>+0.8</b>	-0.6
no TRI encoding	-0.6	-5.8	-3.3	-14.1	-4.4	-9.5

TABLE 3.3: Ablation study for Named Entity Recognition (difference in micro-averaged  $F_1$  score) for different model configurations.

**Named entity annotators.** We distinguish between three broad categories of named entity annotators: *Babelfy* [Moro et al., 2014], *DBpedia Spotlight* [Mendes et al., 2011], *Entityclassifier* [Dojchinovski and Kliegr, 2013] and *TagMe* [Ferragina and Scaiella, 2010] spot noun chunks and filter them with dictionaries, often derived from Wikipedia. *Stanford NER* [Manning et al., 2014] and *LingPipe*<sup>6</sup> utilize discriminative tagging approaches. *FOX* [Speck and Ngomo, 2014] and *NERD-ML* [Van Erp et al., 2013] combine several approaches in an ensemble learner for enhancing precision. *AIDA* [Hoffart et al., 2011] uses a graph-based approach for entity disambiguation.

**Quality measures.** We measure overall performance of all annotators using micro-averaged precision, recall and NER-style  $F_1$  score for exact span match, as defined by Cornolti et al. [2013]. We evaluate these systems in comparison with our TASTY model and utilize the GER-BIL evaluation framework [Usbeck et al., 2015] to run the experiments.

### 3.3.2 Experimental Results

Next, we discuss the evaluation of our TASTY system on NER and NED tasks.

**Named Entity Recognition performance on common English news.** Table 3.1 shows the comparison of TASTY with seven state-of-the-art annotators on common news data sets in English language. We observe that TASTY achieves the highest recall scores of all tested annotators, with 86%–92% on all measured data sets. Most notably, it achieves a state-of-the-art 91.1%  $F_1$  on the CoNLL03-en task. Overall, we achieve high micro- $F_1$  scores of 91%–94% on news entity recognition. We notice that systems specialized on word sense disambiguation (Babelfy, DBpedia Spotlight) don’t perform well on untyped concept recognition and wikification tasks. Stanford NER reaches high precision, but especially lacks recall. Ensemble methods, such as FOX and NERD-ML can help to improve precision drastically, but do not affect recall. We also notice an overall low precision of all annotators on the ACE2004 dataset, which can be explained with differing annotation standards between training and inference time.

<sup>6</sup><http://alias-i.com/lingpipe/>

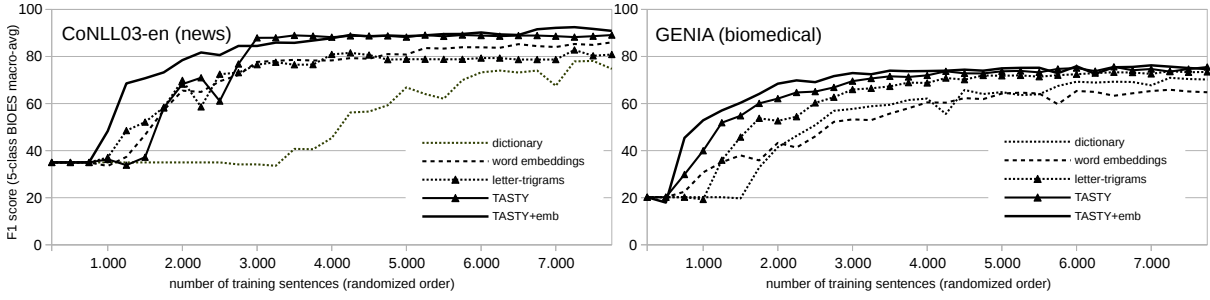


FIGURE 3.3: Effect of the training data size on NER performance ( $F_1$  score) for five different configurations on CoNLL2003-en and GENIA datasets.

**Entity recognition on specific domains.** Table 3.2 shows the application of TASTY to five specialized domains. Across all domains investigated, our system performs equally compared to the English news domain, with micro- $F_1$  scores between 75–88%. Off-the-shelf annotators such as Stanford NER or TagMe do not generalize that well, and even the specialized GENIA model in LingPipe falls short 8.5 percentage points on the corresponding data set. Furthermore, TASTY’s NER performance on German newswire is directly comparable to the results on English text. We achieve 87.9%  $F_1$  on German CoNLL2003 data and 83.3%  $F_1$  on the TIGER dataset. Other annotators do not support German language at all. TASTY’s end-to-end sequence learner is able to adapt to different languages without any hyperparameter changes.

**Ablation study.** Table 3.3 shows the results of experiments on both CoNLL2003 and GENIA data sets with different model configurations. It is clearly visible that the LSTM sequence learning model increases performance significantly compared to feed-forward neural networks (no LSTM). Furthermore, bidirectional LSTM layers contribute another 3.3–3.5 percentage points  $F_1$ . We notice that dictionary-based word encoding (1-hot) works surprisingly well for the medical domain, whereas it suffers from high word ambiguity in the news texts. Using only letter-trigram hashing for word encodings (no EMB) is generally robust, and even improves recall in the medical domain. This domain contains a large number of compound words, so the encoding based on subword-‘syllables’ can provide important information about word meaning. Using only pretrained word2vec embeddings (no TRI) performs well on news data, but cannot adapt to the medical domain without retraining, mainly because of the large number of unseen words. We follow that trigram word encodings are an important factor for robustness and lead to higher precision in special domains and high recall in general.

**Effect of training data size.** Because it is often expensive to obtain labeled data, it is crucial to design effective models that require less training data. As we have shown in the ablation study, TASTY’s architecture supports this scenario with strong generalization and high robustness. Figure 3.3 shows the progress of TASTY’s sequence learner with varying training data sizes. We observe that the TASTY model reaches its peak performance already after training on 4,000–5,000 randomly sampled sentences. For TASTY+emb, the curve is even steeper and further increases with more than 7,000 examples.

Model	MSNBC	ACE2004	AQUAINT	DBpedia Spotlight
Babelfy	<b>70.3</b>	56.5	<b>68.2</b>	51.6
DBpedia Spotlight	38.4	45.6	47.5	60.6
Entityclassifier	–	–	–	25.5
FOX	–	–	42.9	15.3
NERD-ML	–	57.8	59.6	55.6
AIDA	68.4	<b>70.3</b>	55.0	24.9
<b>TASTY+emb</b>	60.8	65.3	61.9	<b>65.4</b>

TABLE 3.4: Experimental results (micro-averaged  $F_1$  score) for Named Entity Disambiguation to English Wikipedia on four standard data sets.

**Disambiguation performance to English Wikipedia.** Table 3.4 shows a comparison of TASTY’s entity disambiguation with six state-of-the-art Entity Linking systems. TASTY achieves highest  $F_1$  scores on the DBpedia data set and is the only NEL system that consistently achieves high scores across all data sets in the range of 61–65%  $F_1$ .

### 3.4 Discussion and Error Analysis

We investigate different aspects of the TASTY components by manual inspection of classification errors in the context of the document. For the error classes (false negative detections, false positives and invalid boundaries), we observe the following causes:

**Unseen words and misspellings.** In dictionary-based configurations (e.g. 1-hot encoding), we observe false negative predictions caused by out-of-vocabulary for words that do not exist in the training data. The cause can be rare, unseen or novel words (e.g. T-prolymphocytic cells) or misspellings (e.g. strengthnend). These words yield a null vector result from the encoder and can therefore not be distinguished by the LSTM. The error increases when using word2vec, because these models are often trained with stop words filtered out, which are very frequent and provide important context. This also implicates that e.g. mentions surrounded by or containing a determiner (e.g. The Sunday Telegraph quoted Majorie Orr) are highly error prone towards the detection of their boundaries. We resolve this error using the letter-trigram approach. Very rare trigrams (e.g. thh) may be missing in the encoding, but this only affect single dimensions as opposed to the vector as a whole.

**Misleading surface form features.** Surface forms encode important features for NER (e.g. capitalization of “new” in Alan Shearer was named as the new England captain and as New York beat the Angels). However, case-sensitive word vectorization methods yield a large amount of false positive predictions caused by incorrect capitalization in the input data. An uppercase headline (e.g. TENNIS - U.S. TEAM ON THE ROAD FOR 1997 FED CUP) is encoded completely different than a lowercase one (e.g. U.S. team on the road for Fed Cup). Because of that, we

achieve best results with lowercase trigram vectors and additional surface form feature flags, as described in Section 3.2.1.

**Syntagmatic and paradigmatic word relations.** We observe mentions that are composed of co-occurring words with high ambiguity (e.g. degradation of IkB alpha in T cell lines). These groups encode strong syntagmatic word relations [Sahlgren, 2008] that can be leveraged to resolve word sense and homonyms from sentence context. Therefore, correct boundaries in these groups can effectively be identified only with contextual models such as LSTMs. Orthogonal to the previous problem, different words in a paradigmatic relation can occur in the same context (e.g. cyclosporin A-treated cells and HU treated cells). These groups are efficiently represented in word2vec. However, letter-trigram vectors cannot encode paradigmatic groups and therefore require a larger training sample to capture these relations.

**Context boundaries.** Often, resolving synonyms requires a larger context than the sentence-level LSTM used in TASTY. In these cases, word sense is often defined by a topic model local to the paragraph (e.g. sports: Tiger was lost in the woods after divorce.). This problem does not heavily affect NER recall, but is crucial for NED performance and coreference resolution.

## 3.5 Conclusions

In this chapter, we have approached RQ 1, the identification of named entities in domain-specific text. We presented TASTY, a robust end-to-end model for Named Entity Recognition and Linking. We have shown that TASTY is able to identify named entities with high recall in domain-specific text and with small available training data. Our end-to-end architecture provides a first step towards general Machine Reading by pushing the costs for creating a new IE model down to labeling a set of training examples. We showed that TASTY is able to recognize generic entity names such as persons, organizations and locations, as well as domain-specific concepts such as disease names, biomedical concepts and car parts. Our model requires labeled training data in the range of 4,000–5,000 sentences to reach an  $F_1$  score above 90%. TASTY leverages contextual information from sentence level using a bidirectional LSTM. It is robust to spelling errors and morphological variations by using a subword-level encoding based on letter trigrams and surface form features<sup>7</sup>. The same features were used to disambiguate the entity names to a knowledge base without additional model training. We further included corpus-level background knowledge by including distributed word embeddings. This helped to further speed up the training process and solved generalization errors emerging from lack of available training data. Finally, one aspect of RQ 1 still remains open: can contextual information on document level, especially structure and topic information, further improve the identification of named entities? We will come back to this question in Chapter 5.

<sup>7</sup>After this chapter was published, related research proposed a combination of letter n-gram based and distributional approaches for word encoding [Bojanowski et al., 2017]. The *Fasttext* model comprises most of the advantages for robust word encoding presented in this chapter and we will use it in the following chapters.

## Chapter 4

# Coherent Topic Segmentation and Classification

When searching for information, a human reader first glances over a document, spots relevant sections and then focuses on a few sentences for resolving her intention. To reproduce this process, a Machine Reading system requires to identify the structure of a document and highlight the salient topics for each section. In this chapter, we approach RQ 2: *How can Machine Reading models detect topics and structure in long documents?* We propose the task of segmenting long documents into coherent sections and assigning topic labels to each section.

We present SECTOR, a Neural Machine Reading model which learns a latent topic embedding over the course of a document<sup>1</sup>. Our model utilizes bidirectional LSTMs with Bloom filter embeddings on sentence level. We apply SECTOR to the task of coherent topic segmentation and classification into up to 30 topics. We introduce the WIKISECTION dataset, which contains long documents in English and German labeled for this task from two distinct domains: medicine and geopolitics. Additionally, we evaluate SECTOR’s performance on four English datasets from clinical medicine, biomedicine, geopolitics and general encyclopedia. We show that SECTOR performs classification with high accuracy across all domains and can adapt to various datasets to predict boundaries of coherent passages.

This chapter is structured as follows: In Section 4.1, we define the task of topic segmentation and classification. In Section 4.2, we introduce our WIKISECTION dataset for this task. In Section 4.3, we present the architecture for our SECTOR model. We formulate three different sentence representations as input features (Section 4.3.1). We describe SECTOR’s central topic embedding which is trained for single and multi-label multi-class topic classification (Section 4.3.2). We introduce our embedding deviation method for segmenting topics (Section 4.3.3). In Section 4.4, we evaluate different configurations of our SECTOR model in comparison to 12 architectures from related work. In Section 4.5, we discuss the results and give insights into the internal representations of SECTOR. In Section 4.6, we discuss related work. We summarize this chapter in Section 4.7 and point out important building blocks for our vision of Neural Machine Reading.

---

<sup>1</sup>This chapter was published by S. Arnold, R. Schneider, P. Cudré-Mauroux, F. A. Gers, and A. Löser [2019]. “SECTOR: A Neural Model for Coherent Topic Segmentation and Classification”. In: *Transactions of the Association for Computational Linguistics* 7, pp. 169–184 with oral presentation at ACL’2019.

## 4.1 Introduction

Today’s systems for natural language understanding are comprised of building blocks that extract semantic information from the text, such as named entities, relations, topics or discourse structure. In traditional Natural Language Processing (NLP), these extractors are typically applied to bags of words or full sentences [Hirschberg and Manning, 2015]. Recent neural architectures build upon pre-trained word or sentence embeddings [Mikolov et al., 2013a; Le and Mikolov, 2014], which focus on semantic relations that can be learned from large sets of paradigmatic examples, even from long ranges [Dieng et al., 2017].

From a human perspective, however, it is mostly the authors themselves who help best to understand a text. Especially in long documents, an author thoughtfully designs a readable structure and guides the reader through the text by arranging topics into coherent passages [Glavaš et al., 2016]. In many cases, this structure is not formally expressed as section headings (e.g. in news articles, reviews, discussion forums) or it is structured according to domain-specific aspects (e.g. health reports, research papers, insurance documents).

### 4.1.1 Challenges for Topic Representation

Ideally, systems for text analytics, such as Topic Detection and Tracking (TDT) [Allan, 2002], text summarization [Huang et al., 2003], Information Retrieval (IR) [Dias et al., 2007] or Question Answering (QA) [Cohen et al., 2018] could access a document representation that is aware of both *topical* (i.e. latent semantic content) and *structural* information (i.e. segmentation) in the text [MacAvaney et al., 2018]. The challenge in building such a representation is to combine these two dimensions which are strongly interwoven in the author’s mind. It is therefore important to understand topic segmentation and classification as a mutual task that requires to encode both topic information and document structure coherently.

In this chapter, we present SECTOR<sup>2</sup>, an end-to-end model which learns an embedding of latent topics from potentially ambiguous headings and can be applied to entire documents to predict local topics on sentence level. Our model encodes topical information on a vertical dimension and structural information on a horizontal dimension. We show that the resulting embedding can be leveraged in a downstream pipeline to segment a document into coherent sections and classify the sections into one of up to 30 topic categories reaching 71.6%  $F_1$  – or alternatively attach up to 603 topic labels with 71.1% MAP. We further show that segmentation performance of our bidirectional LSTM architecture is comparable to specialized state-of-the-art segmentation methods on various real-world datasets.

To the best of our knowledge, the combined task of segmentation and classification has not been approached on full document level before. There exist a large number of datasets for text segmentation, but most of them do not reflect real-world topic drifts [Choi, 2000; Sehikh et al., 2017], do not include topic labels [Eisenstein and Barzilay, 2008; Jeong and Titov, 2010; Glavaš

<sup>2</sup>Our source code is available under the Apache License 2.0 at <https://github.com/sebastianarnold/SECTOR>



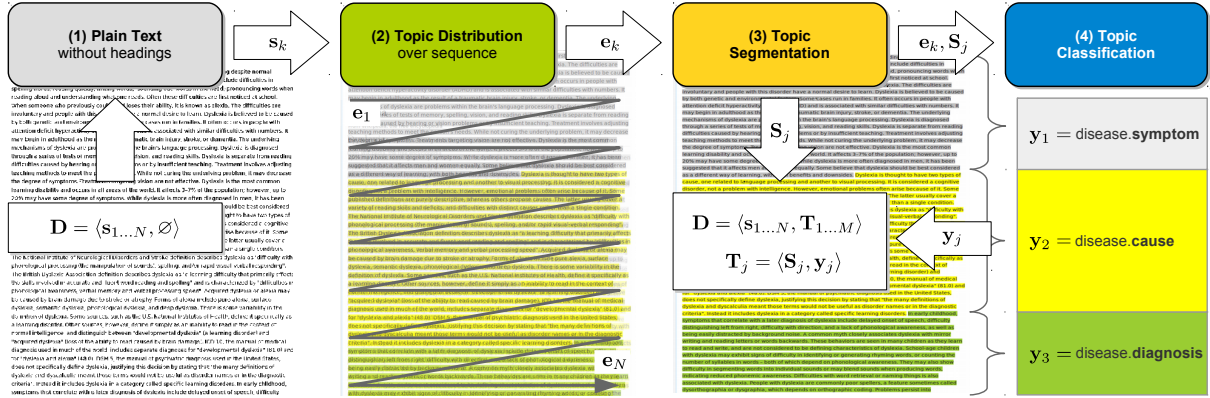


FIGURE 4.1: Overview of the WIKISECTION task: (1) The input is a plain text document  $D$  without structure information. (2) We assume the sentences  $s_{1...N}$  contain a coherent sequence of local topics  $e_{1...N}$ . (3) The task is to segment the document into coherent sections  $S_{1...M}$  and (4) to classify each section with a topic label  $y_{1...M}$ .

et al., 2016] or are heavily normalized and too small to be used for training neural networks [Chen et al., 2009]. We can utilize a generic segmentation dataset derived from Wikipedia that includes headings [Koshorek et al., 2018], but there is also a need in IR and QA for supervised structural topic labels [Agarwal and Yu, 2009; MacAvaney et al., 2018], different languages and more specific domains, such as clinical or biomedical research [Tepper et al., 2012; Tsatsaronis et al., 2012] and news-based TDT [Kumaran and Allan, 2004; Leetaru and Schrodt, 2013].

Therefore we introduce WIKISECTION<sup>3</sup>, a large novel dataset of 38K articles from the English and German Wikipedia labeled with 242K sections, original headings and normalized topic labels for up to 30 topics from two domains: *diseases* and *cities*. We chose these subsets to cover both clinical/biomedical aspects (e.g. symptoms, treatments, complications) and news-based topics (e.g. history, politics, economy, climate). Both article types are reasonably well-structured according to Wikipedia guidelines [Piccardi et al., 2018], but we show that they are also complementary: diseases is a typical scientific domain with low entropy, i.e. very narrow topics, precise language and low word ambiguity. In contrast, cities resembles a diversified domain, with high entropy, i.e. broader topics, common language and higher word ambiguity, and will be more applicable to e.g. news, risk reports or travel reviews.

We compare SECTOR to existing segmentation and classification methods based on latent dirichlet allocation (LDA), paragraph embeddings, convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We show that SECTOR significantly improves these methods in a combined task by up to 29.5 points  $F_1$  when applied to plain text with no given segmentation.

<sup>3</sup>The dataset is available under the CC BY-SA 3.0 license at <https://github.com/sebastianarnold/WikiSection>

Dataset	disease		city	
	en	de	en	de
language				
total docs	3.6K	2.3K	19.5K	12.5K
avg sents per doc	58.5	45.7	56.5	39.9
avg sects per doc	7.5	7.2	8.3	7.6
headings	8.5K	6.1K	23.0K	12.2K
topics	27	25	30	27
coverage	94.6%	89.5%	96.6%	96.1%

TABLE 4.1: Dataset characteristics for *disease* (German: *Krankheit*) and *city* (German: *Stadt*). *Headings* denotes the number of distinct section and subsection headings among the documents. *Topics* stands for the number of topic labels after synset clustering. *Coverage* denotes the proportion of headings covered by topics; the remaining headings are labeled as other.

### 4.1.2 Task Definition

We start with a definition of the WIKISECTION Machine Reading task shown in Figure 4.1. We take a document  $D = \langle \mathcal{S}, \mathcal{T} \rangle$  consisting of  $N$  consecutive sentences  $\mathcal{S} = (s_1, \dots, s_N)$  and empty segmentation  $\mathcal{T} = \emptyset$  as input. In our example, this is the plain text of a Wikipedia article (e.g. about Trichomoniasis<sup>4</sup>) without any section information. For each sentence  $s_k$ , we assume a distribution of local topics  $\mathbf{e}_k$  that gradually changes over the course of the document.

The task is to split  $D$  into a sequence of distinct topic sections  $\mathcal{T} = (T_1, \dots, T_M)$ , so that each predicted section  $T_j = \langle \mathcal{S}_j, y_j \rangle$  contains a sequence of coherent sentences  $\mathcal{S}_j \subseteq \mathcal{S}$  and a topic label  $y_j$  that describes the common topic in these sentences. For the document Trichomoniasis, the sequence of topic labels is  $y_{1..M} = (\text{symptom, cause, diagnosis, prevention, treatment, complication, epidemiology})$ .

## 4.2 WikiSection Dataset

For the evaluation of this task, we created WIKISECTION, a novel dataset containing a gold standard of 38K full-text documents from English and German Wikipedia comprehensively annotated with sections and topic labels (see Table 4.1).

The documents originate from recent dumps in English<sup>5</sup> and German<sup>6</sup>. We filtered the collection using SPARQL queries against Wikidata [Tanon et al., 2016]. We retrieved instances of Wikidata categories *disease* (Q12136) and their subcategories, e.g. Trichomoniasis or Pertussis, or *city* (Q515), e.g. London or Madrid.

Our dataset contains the article abstracts, plain text of the body, positions of all sections given by the Wikipedia editors with their original headings (e.g. “Causes | Genetic sequence”) and a normalized topic label (e.g. disease.cause). We randomized the order of documents and split them into 70% training, 10% validation, 20% test sets.

<sup>4</sup><https://en.wikipedia.org/w/index.php?title=Trichomoniasis&oldid=814235024>

<sup>5</sup><https://dumps.wikimedia.org/enwiki/20180101>

<sup>6</sup><https://dumps.wikimedia.org/dewiki/20180101>

rank	heading $h$	label $y$	$H$	freq
0	Diagnosis	diagnosis	0.68	3,854
1	Treatment	treatment	0.69	3,501
2	Signs and Symptoms	symptom	0.68	2,452
...				
21	Differential Diagnosis	diagnosis	0.23	236
22	Pathogenesis	mechanism	0.16	205
23	Medications	medication	0.14	186
...				
8,494	Usher Syndrome Type IV	classification	0.00	1
8,495	False Melanose Lesions	other	0.00	1
8,496	Cognitive Therapy	treatment	0.00	1

TABLE 4.2: Frequency and entropy ( $H$ ) of top-3 head and randomly selected torso and tail headings for category diseases in the English Wikipedia.

### 4.2.1 Preprocessing

To obtain plain document text, we used Wikiextractor<sup>7</sup>, split the abstract sections and stripped all section headings and other structure tags except newline characters and lists.

**Vocabulary mismatch in section headings.** Table 4.2 shows examples of section headings from disease articles separated into head (most common), torso (frequently used) and tail (rare). Initially, we expected articles to share congruent structure in naming and order. Instead, we observe a high variance with 8.5K distinct headings in the diseases domain and over 23K for English cities. A closer inspection reveals that Wikipedia authors utilize headings at different granularity levels, frequently copy and paste from other articles, but also introduce synonyms or hyponyms, which leads to a *vocabulary mismatch problem* [Furnas et al., 1987]. As a result, the distribution of headings is heavy-tailed across all articles. Roughly 1% of headings appear more than 25 times while the vast majority (88%) appear 1 or 2 times only.

### 4.2.2 Synset Clustering

In order to use Wikipedia headlines as a source for topic labels, we contribute a normalization method to reduce the high variance of headings to few representative labels based on the clustering of BabelNet synsets [Navigli and Ponzetto, 2012].

We create a set  $\mathcal{H}$  that contains all headings in the dataset and use the BabelNet API to match<sup>8</sup> each heading  $h \in \mathcal{H}$  to its corresponding synsets  $S_h \subset S$ . For example, “Cognitive behavioral therapy” is assigned to synset bn:03387773n. Next, we insert all matched synsets into

<sup>7</sup><http://attardi.github.io/wikiextractor/>

<sup>8</sup>We match lemmas of main senses and compounds to synsets of type NOUN CONCEPT.

en_disease (27)	de_disease (25)	en_city (30)	de_city (27)
cause	definition	architecture	architektur
classification	diagnose	climate	bildung
complication	epidemiologie	crime	demografie
culture	fauna	culture	erholung
diagnosis	forschung	demography	etymologie
epidemiology	genetik	district	gemeinde
etymology	geographie	economics	gemeindeparterschaft
fauna	geschichte	education	geographie
genetics	infektion	environment	geschichte
geography	kategorisierung	etymology	infrastruktur
history	klinik	facility	kirche
infection	komplikation	faith	klima
management	mensch	geography	kriminalität
mechanism	organe	health	kultur
medication	pathologie	history	menschen
pathology	prävalenz	infrastructure	politik
pathophysiology	prognose	international_affairs	presse
prevention	risiko	law	regierung
prognosis	symptom	media	religion
research	terminologie	overview	sport
risk	therapie	people	stadtlandschaft
screening	ursache	politics	stadtviertel
surgery	verlauf	recreation	tourismus
symptom	vorbeugung	science	überblick
tomography	sonstiges	sights	verkehr
treatment		society	wirtschaft
other		sport	sonstiges
		tourism	
		transport	
		other	

TABLE 4.3: List of topics contained in the four WIKISECTION datasets (representative cluster labels in alphabetical order).

an undirected graph  $G$  with nodes  $s \in S$  and edges  $e$ . We create edges between all synsets that match among each other with a lemma  $h' \in \mathcal{H}$ . Finally, we apply a community detection algorithm [Newman, 2006] on  $G$  to find dense clusters of synsets. We use these clusters as normalized topics and assign the sense with most outgoing edges as representative label, in our example e.g. therapy.

From this normalization step we obtain 598 synsets which we prune using the head/tail division rule  $\text{count}(s) < \frac{1}{|S|} \sum_{s_i \in S} \text{count}(s_i)$  [Jiang, 2012]. This method covers over 94% of all headings and yields 26 normalized labels and one other class in the English disease dataset. Table 4.1 shows the corresponding numbers for the other datasets, a full list of topics is shown in Table 4.3. We verify our normalization process by manual inspection of 400 randomly chosen heading-label assignments by two independent judges and report an accuracy of 97.2% with an average observed inter-annotator agreement of 96.0%.

### 4.3 Model Architecture

We introduce SECTOR, a neural embedding model that predicts a latent topic distribution for every position in a document. Based on the task described the previous section, we aim to detect  $M$  sections  $\mathcal{T}_{1\dots M}$  in a document  $D$  and assign topic labels  $y_j = \text{topic}(\mathcal{S}_j)$ , where  $j = 1, \dots, M$ . Because we do not know the expected number of sections, we formulate the objective of our model on sentence level and later segment based on the predictions. Therefore, we assign each sentence  $s_k$  a sentence topic label  $\bar{y}_k = \text{topic}(s_k)$ , where  $k = 1, \dots, N$ . Thus, we aim to predict coherent sections with respect to document context:

$$p(\bar{y}_1, \dots, \bar{y}_N \mid D) = \prod_{k=1}^N p(\bar{y}_k \mid s_1, \dots, s_N) \quad (4.1)$$

We approach two variations of this task: for WIKISECTION-topics, we choose a single topic label  $y_j \in \mathcal{Y}$  out of a small number of normalized topic labels. However, from this simplified classification task arises an entailment problem, because topics might be hierarchically structured. For example, a section with heading ‘‘Treatment | Gene Therapy’’ might describe genetics as a subtopic of treatment. Therefore, we also approach an extended task WIKISECTION-headings to capture ambiguity in a heading. We follow the CBOW approach [Mikolov et al., 2013a] and assign all words in the heading  $z_j \subset \mathcal{Z}$  as multi-label bag over the original heading vocabulary. This turns our problem into a ranked retrieval task with a large number of ambiguous labels, similar to Prabhu and Varma [2014]. It further eliminates the need for normalized topic labels. Because this ‘noisy’ task is much larger and more ambiguous, our approach is similar to extreme multi-label text classification [Prabhu and Varma, 2014]. For both tasks, we aim to maximize the log likelihood of model parameters  $\Theta$  per section and sentence:

$$\begin{aligned} \mathcal{L}(\Theta)_{\text{sect}} &= \sum_{j=1}^M \log p(y_j \mid s_1, \dots, s_N; \Theta) \\ \bar{\mathcal{L}}(\Theta)_{\text{sent}} &= \sum_{k=1}^N \log p(\bar{y}_k \mid s_1, \dots, s_N; \Theta) \end{aligned} \quad (4.2)$$

Our SECTOR architecture consists of four stages shown in Figure 4.2: sentence encoding, topic embedding, topic classification and topic segmentation. We discuss each stage in the following sections.

#### 4.3.1 Sentence Representation

The first stage of our SECTOR model transforms each sentence  $s_k$  from plain text into a fixed-size sentence vector  $\mathbf{x}_k$  which serves as input into the neural network layers. Following Hill et al. [2016], word order is not critical for document-centric evaluation settings. Therefore, we mainly focus on unsupervised compositional sentence representations.

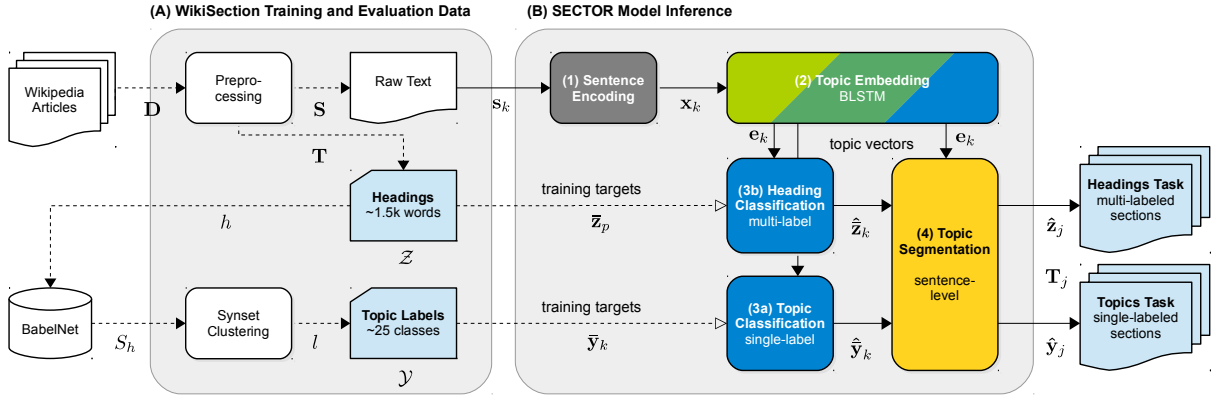


FIGURE 4.2: Training and inference phases of segmentation and topic classification (SECTOR). For training (A), we preprocess Wikipedia documents to supply a ground truth for segmentation  $T$ , headings  $Z$  and topic labels  $Y$ . During inference (B), we invoke SECTOR with unseen plain text to predict topic embeddings  $e_k$  on sentence level. The embeddings are used to segment the document and classify headings  $\hat{z}_j$  and normalized topic labels  $\hat{y}_j$ .

**Bag-of-words baseline.** As a baseline, we compose sentence vectors using a weighted bag-of-words scheme. Let  $e_w \in \{0, 1\}^{|\mathcal{V}|}$  be the indicator vector, such that  $e_w^{(i)} = 1$  iff  $w$  is the  $i$ -th word in the fixed vocabulary  $\mathcal{V}$ , and let  $\text{TF-IDF}(w)$  be the TF-IDF weight of  $w$  in the corpus. We define the sparse bag-of-words encoding  $\mathbf{x}_{\text{bow}} \in \mathbb{R}^{|\mathcal{V}|}$  as follows:

$$\mathbf{x}_{\text{bow}}(s) = \sum_{w \in s} (\text{TF-IDF}(w) \cdot e_w) \quad (4.3)$$

**Bloom filter encoding.** For large  $\mathcal{V}$  and long documents, input matrices grow too large to fit into GPU memory, especially with larger batch sizes. Therefore we apply a compression technique for sparse sentence vectors based on Bloom filters [Serrà and Karatzoglou, 2017]. A Bloom filter projects every item of a set onto a bit array  $\mathbb{A}(i) \in \{0, 1\}^m$  using  $k$  independent hash functions. We use the sum of bit arrays per word as compressed Bloom embedding  $\mathbf{x}_{\text{bloom}} \in \mathbb{N}^m$ :

$$\mathbf{x}_{\text{bloom}}(s) = \sum_{w \in s} \sum_{i=1}^k \mathbb{A}(\text{hash}_i(w)) \quad (4.4)$$

We set parameters to  $m = 4096$  and  $k = 5$  to achieve a compression factor of 0.2, which showed good performance in the original paper.

**Sentence embeddings.** We use the strategy of Arora et al. [2017] to generate a distributed sentence representation based on pre-trained word2vec embeddings [Mikolov et al., 2013a]. This method composes a sentence vector  $\mathbf{v}_{\text{emb}} \in \mathbb{R}^d$  for all sentences using a probability-weighted sum of word embeddings  $\mathbf{v}_w \in \mathbb{R}^d$  with  $\alpha = 10^{-4}$  and subtracts the first principal component

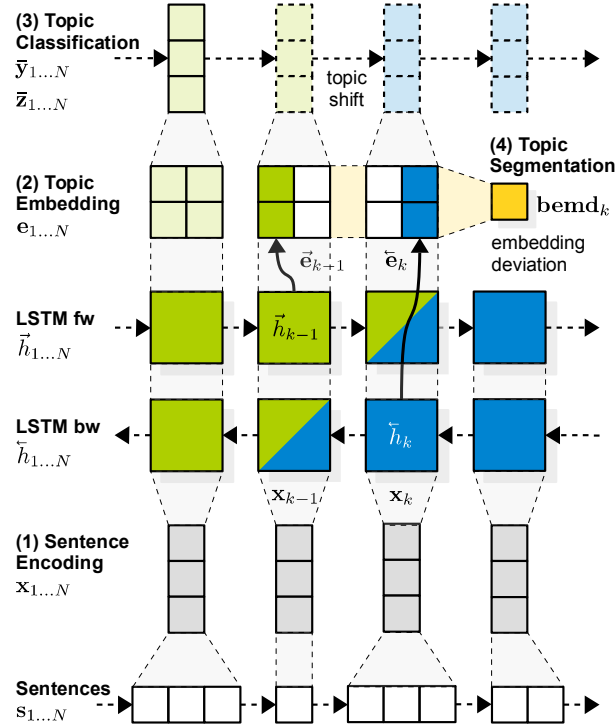


FIGURE 4.3: Neural network architecture SECTOR. The recurrent model consists of stacked LSTM, embedding and output layers that are optimized on document level and later accessed during inference in stages 1–4.

u of the embedding matrix  $[\mathbf{v}_s : s \in \mathcal{S}]$ :

$$\mathbf{v}_s = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \left( \frac{\alpha}{\alpha + p(w)} \mathbf{v}_w \right) \quad (4.5)$$

$$\mathbf{x}_{\text{emb}}(s) = \mathbf{v}_s - \mathbf{u}\mathbf{u}^\top \mathbf{v}_s$$

### 4.3.2 Topic Classification

We model the second stage in our architecture to produce a dense distributed representation of latent topics for each sentence in the document.

**Sequential topic embedding.** We use two layers of LSTM [Hochreiter and Schmidhuber, 1997] with forget gates [Gers et al., 2000] connected to read the document in forward and backward direction [Graves, 2012]. We feed the LSTM outputs to a ‘bottleneck’ layer with tanh activation as topic embedding. Figure 4.3 shows these layers in context of the complete architecture. We can see that context from left  $(k - 1)$  and right  $(k + 1)$  affects forward and backward layers independently. It is therefore important to separate these weights in the embedding layer to precisely capture the difference between sentences at section boundaries. We modify our objective given in Eq. 4.2 accordingly with long-range dependencies from forward and backward

layers of the LSTM:

$$\begin{aligned} \mathcal{L}(\Theta) = \sum_{k=1}^N & (\log p(\bar{\mathbf{y}}_k \mid \mathbf{x}_{1\dots k-1}; \vec{\Theta}, \Theta')) \\ & + \log p(\bar{\mathbf{y}}_k \mid \mathbf{x}_{k+1\dots N}; \vec{\Theta}, \Theta')) \end{aligned} \quad (4.6)$$

Note that we separate network parameters  $\vec{\Theta}$  and  $\vec{\Theta}$  for forward and backward directions of the LSTM, and tie the remaining parameters  $\Theta'$  for the embedding and output layers. This strategy couples the optimization of both directions into the same vector space without the need for an additional loss function. The embeddings  $\mathbf{e}_{1\dots N}$  are calculated from the context-adjusted hidden states  $h'_k$  of the LSTM cells (here simplified as  $f_{\text{LSTM}}$ ) through the bottleneck layer:

$$\begin{aligned} \vec{h}_k &= f_{\text{LSTM}}(\mathbf{x}_k, \vec{h}'_{k-1}, \vec{\Theta}) \\ \vec{h}_k &= f_{\text{LSTM}}(\mathbf{x}_k, \vec{h}'_{k+1}, \vec{\Theta}) \\ \vec{\mathbf{e}}_k &= \tanh(W_{eh}\vec{h}_k + b_e) \\ \vec{\mathbf{e}}_k &= \tanh(W_{eh}\vec{h}_k + b_e) \end{aligned} \quad (4.7)$$

Now, a simple concatenation of the embeddings  $\mathbf{e}_k = \vec{\mathbf{e}}_k \oplus \vec{\mathbf{e}}_k$  can be used as topic vector by downstream applications.

**Single-class topic classification.** The third stage in our architecture is the output layer that decodes the class labels. To learn model parameters  $\Theta$  required by the embedding, we need to optimize the full model for a training target. For the WIKISECTION-topics task, we use a simple one-hot encoding  $\bar{\mathbf{y}} \in \{0, 1\}^{|\mathcal{Y}|}$  of the topic labels constructed in Section 4.1.2 with a softmax activation output layer. For the WIKISECTION-headings task, we encode each heading as lowercase bag-of-words vector  $\bar{\mathbf{z}} \in \{0, 1\}^{|\mathcal{Z}|}$ , such that  $\bar{\mathbf{z}}^{(i)} = 1$  iff the  $i$ -th word in  $\mathcal{Z}$  is contained in the heading, e.g.  $\bar{\mathbf{z}}_k \hat{=} \{\text{gene, therapy, treatment}\}$ . We then use a sigmoid activation function:

$$\begin{aligned} \hat{\mathbf{y}}_k &= \text{softmax}(\mathbf{W}_{ye}\vec{\mathbf{e}}_k + \mathbf{W}_{ye}\vec{\mathbf{e}}_k + \mathbf{b}_y) \\ \hat{\mathbf{z}}_k &= \text{sigmoid}(\mathbf{W}_{ze}\vec{\mathbf{e}}_k + \mathbf{W}_{ze}\vec{\mathbf{e}}_k + \mathbf{b}_z) \end{aligned} \quad (4.8)$$

**Multi-class topic classification.** The multi-label objective is to maximize the likelihood of every word that appears in a heading:

$$\mathcal{L}(\Theta) = \sum_{k=1}^N \sum_{i=1}^{|\mathcal{Z}|} \log p(\bar{\mathbf{z}}_k^{(i)} \mid \mathbf{x}_{1\dots N}; \Theta) \quad (4.9)$$

**Ranking loss.** For training this model, we use a variation of the logistic pairwise ranking loss function proposed by Santos et al. [2015]. It learns to maximize the distance between positive and negative labels:

$$\begin{aligned} \mathcal{L}_{\text{rank}} &= \log(1 + \exp(\gamma(m^+ - \text{score}^+(\mathbf{x})))) \\ &+ \log(1 + \exp(\gamma(m^- + \text{score}^-(\mathbf{x})))) \end{aligned} \quad (4.10)$$



We calculate the positive term of the loss by taking all scores of correct labels  $y^+$  into account. We average over all correct scores to avoid a too strong positive push on the energy surface of the loss function [LeCun et al., 2006]. For the negative term, we only take the most offending example  $y^-$  among all incorrect class labels.

$$\begin{aligned}\text{score}^+(\mathbf{x}) &= \frac{1}{|y^+|} \sum_{y \in y^+} s_\theta(\mathbf{x})^{(y)} \\ \text{score}^-(\mathbf{x}) &= \arg \max_{y \in y^-} s_\theta(\mathbf{x})^{(y)}\end{aligned}\tag{4.11}$$

Here,  $s_\theta(\mathbf{x})^{(y)}$  denotes the score of label  $y$  for input  $\mathbf{x}$ . We follow the authors and set scaling factor  $\gamma = 2$ , margins  $m^+ = 2.5$  and  $m^- = 0.5$ .

### 4.3.3 Topic Segmentation

In the final stage, we leverage the information encoded in the topic embedding and output layers to segment the document and classify each section.

**Baseline segmentation methods.** As a simple baseline method, we use prior information from the text and split sections at *newline* characters (NL). Additionally, we merge two adjacent sections if they are assigned the same topic label after classification. If there is no newline information available in the text, we use a *maximum label* (max) approach: We first split sections at every sentence break, i.e.  $S_j = s_k; j = k = 1, \dots, N$  and then merge all sections which share at least one label in the top-2 predictions.

**Embedding deviation.** All information required to classify each sentence in a document is contained in our dense topic embedding matrix  $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_N]$ . We are now interested in the vector space movement of this embedding over the sequence of sentences. Therefore, we apply a number of transformations adapted from Laplacian-of-Gaussian edge detection on images [Ziou and Tabbone, 1998] to obtain the magnitude of *embedding deviation* (emd) per sentence. First, we reduce the dimensionality of  $\mathbf{E}$  to  $D$  dimensions using PCA, i.e. we solve  $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^\top$  using singular value decomposition and then project  $\mathbf{E}$  on the  $D$  principal components  $\mathbf{E}_D = \mathbf{E}\mathbf{W}_D$ . Next, we apply Gaussian smoothing to obtain a smoothed matrix  $\mathbf{E}'_D$  by convolution with a Gaussian kernel with variance  $\sigma^2$ . From the reduced and smoothed embedding vectors  $\mathbf{e}'_{1\dots N}$  we construct a sequence of deviations  $\mathbf{d}_{1\dots N}$  by calculating the stepwise difference using cosine distance:

$$\mathbf{d}_k = \cos(\mathbf{e}'_{k-1}, \mathbf{e}'_k) = \frac{\mathbf{e}'_{k-1} \cdot \mathbf{e}'_k}{\|\mathbf{e}'_{k-1}\| \|\mathbf{e}'_k\|}\tag{4.12}$$

Finally we apply the sequence  $\mathbf{d}_{1\dots N}$  with parameters  $D = 16$  and  $\sigma = 2.5$  to locate the spots of fastest movement (see Figure 4.4), i.e. all  $k$  where  $\mathbf{d}_{k-1} < \mathbf{d}_k > \mathbf{d}_{k+1}; k = 1 \dots N$  in our discrete case. We use these positions to start a new section.

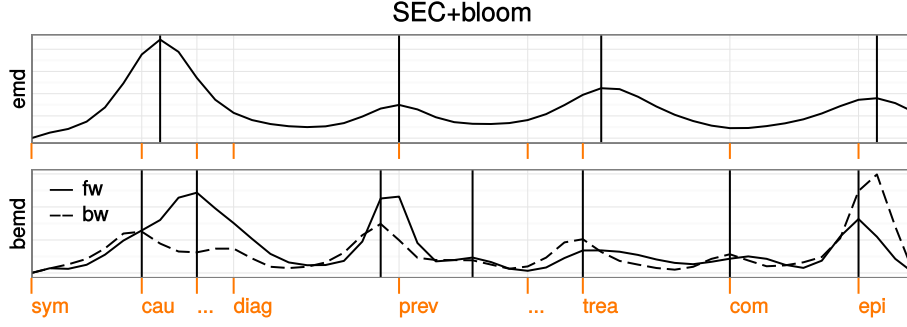


FIGURE 4.4: Embedding deviations  $\text{emd}_k$  and  $\text{bemd}_k$  of the smoothed SECTOR topic embeddings for example document Trichomoniasis. The plot shows the first derivative of vector movement over sentences  $k = 1, \dots, N$  from left to right. Predicted segmentation is shown as black lines, the axis labels indicate ground truth segmentation.

**Bidirectional embedding deviation.** We adopt the approach of Seikh et al. [2017], who examine the difference between forward and backward layer of an LSTM for segmentation. However, our approach focuses on the difference of left and right topic context over time steps  $k$ , which allows for a sharper distinction between sections. Here, we obtain two smoothed embeddings  $\vec{e}$  and  $\tilde{\vec{e}}$  and define the *bidirectional embedding deviation* ( $\text{bemd}$ ) as geometric mean of the forward and backward difference:

$$\mathbf{d}'_k = \sqrt{\cos(\vec{e}_{k-1}, \vec{e}_k) \cdot \cos(\tilde{\vec{e}}_k, \tilde{\vec{e}}_{k+1})} \quad (4.13)$$

Finally, we assign each segment the mean class distribution of all contained sentences:

$$\hat{\mathbf{y}}_j = \frac{1}{|\mathcal{S}_j|} \sum_{s_i \in \mathcal{S}_j} \hat{\mathbf{y}}_i \quad (4.14)$$

We show in the evaluation that our SECTOR model which was optimized for sentences  $\bar{y}_k$  can be applied to the WIKISECTION task to predict coherently labeled sections  $\mathcal{T}_j = \langle \mathcal{S}_j, \hat{\mathbf{y}}_j \rangle$ .

## 4.4 Evaluation

We conduct three experiments to evaluate the segmentation and classification task introduced in Section 4.1.2. The WIKISECTION-topics experiment comprises segmentation and classification of each section with a single topic label out of a small number of clean labels (25–30 topics). The WIKISECTION-headings experiment extends the classification task to multi-label per section with a larger target vocabulary (115–601 words<sup>9</sup>). This is important, because often there are no clean topic labels available for training or evaluation. Finally, we conduct a third experiment to see how SECTOR performs across existing segmentation datasets.

<sup>9</sup>The original vocabulary of 1.0K–2.8K words reported in the paper was pruned during preprocessing

#### 4.4.1 Evaluation Set-up

**Evaluation datasets.** For the first two experiments we use the WIKISECTION datasets introduced in Section 4.2, which contain documents about diseases and cities in both English and German. The subsections are retained with full granularity. For the third experiment, text segmentation results are often reported on artificial datasets [Choi, 2000]. It was shown that this scenario is hardly applicable to topic-based segmentation [Koshorek et al., 2018], so we restrict our evaluation to real-world datasets that are publicly available. The *Wiki-727k* dataset by Koshorek et al. [2018] contains Wikipedia articles with a broad range topics and their top-level sections. However, it is too large to compare exhaustively, so we use the smaller *Wiki-50* subset. We further use *Cities* and *Elements* datasets introduced by Chen et al. [2009], which also provide headings. These sets are typically used for word-level segmentation, so they don’t contain any punctuation and are lowercased. Finally, we use the *Clinical Textbook* chapters introduced by Eisenstein and Barzilay [2008], which do not supply headings.

**Text segmentation models.** We compare SECTOR to baseline text segmentation methods, *C99* [Choi, 2000] and *TopicTiling* [Riedl and Biemann, 2012] and the state-of-the-art *TextSeg* [Koshorek et al., 2018]. In the third experiment we report numbers for *BayesSeg* [Eisenstein and Barzilay, 2008] (without given number of segments) and *GraphSeg* [Glavaš et al., 2016].

**Classification models.** We compare SECTOR to existing models for single and multi-label sentence classification. Because we are not aware of any existing method for combined segmentation and classification, we first use given prior segmentation from newlines in the text (*NL*) and then additionally apply our own segmentation strategies for plain text input: maximum label (*max*), embedding deviation (*emd*) and bidirectional embedding deviation (*bemd*).

For the experiments, we train a Paragraph Vectors (*PV*) model [Le and Mikolov, 2014] using all sections of the training sets. We utilize this model for single-label topic classification (depicted as *PV>T*) by assigning the given topic labels as paragraph IDs. Multi-label classification is not possible with this model. We use the paragraph embedding for our own segmentation strategies. We set the layer size to 256, window size to 7 and trained for 10 epochs using a batch size of 512 sentences and a learning rate of 0.025. We further use an implementation of *CNN* [Kim, 2014] with our pre-trained word vectors as input for single-label topics (*CNN>T*) and multi-label headings (*CNN>H*). We configured the models using the hyperparameters given in the paper and trained the model using a batch size of 256 sentences for 20 epochs with learning rate 0.01.

**SECTOR configurations.** We evaluate the various configurations of our model discussed in prior sections. *SEC>T* depicts the single-label topic classification model which uses a softmax activation output layer, *SEC>H* is the multi-label variant with a larger output and sigmoid activations. Other options are: bag-of-words sentence encoding (*+bow*), Bloom filter encoding (*+bloom*) and sentence embeddings (*+emb*); multi-class cross-entropy loss (as default)

WikiSection-topics single-label classification		en_disease 27 topics			de_disease 25 topics			en_city 30 topics			de_city 27 topics		
model configuration	segm.	$P_k$	$F_1$	MAP	$P_k$	$F_1$	MAP	$P_k$	$F_1$	MAP	$P_k$	$F_1$	MAP
<b>Classification with newline prior segmentation</b>													
PV>T*	NL	35.6	31.7	47.2	36.0	29.6	44.5	22.5	52.9	63.9	27.2	42.9	55.5
CNN>T*	NL	31.5	40.4	55.6	31.6	38.1	53.7	13.2	66.3	76.1	13.7	63.4	75.0
SEC>T+bow	NL	25.8	54.7	68.4	25.0	<b>52.7</b>	<b>66.9</b>	21.0	43.7	55.3	20.2	40.5	52.2
SEC>T+bloom	NL	22.7	<b>59.3</b>	<b>71.9</b>	27.9	50.2	65.5	<b>9.8</b>	<b>74.9</b>	<b>82.6</b>	11.7	73.1	81.5
SEC>T+emb*	NL	<b>22.5</b>	58.7	71.4	<b>23.6</b>	50.9	66.8	10.7	74.1	82.2	<b>10.7</b>	<b>74.0</b>	<b>83.0</b>
<b>Classification and segmentation on plain text</b>													
C99		37.4	n/a	n/a	42.7	n/a	n/a	36.8	n/a	n/a	38.3	n/a	n/a
TopicTiling		43.4	n/a	n/a	45.4	n/a	n/a	30.5	n/a	n/a	41.3	n/a	n/a
TextSeg		<b>24.3</b>	n/a	n/a	35.7	n/a	n/a	19.3	n/a	n/a	27.5	n/a	n/a
PV>T*	max	43.6	20.4	36.5	44.3	19.3	34.6	31.1	28.1	43.1	36.4	20.2	35.5
PV>T*	emd	39.2	32.9	49.3	37.4	32.9	48.7	24.9	53.1	65.1	32.9	40.6	55.0
CNN>T*	max	40.1	26.9	45.0	40.7	25.2	43.8	21.9	42.1	58.7	21.4	42.1	59.5
SEC>T+bow	max	30.1	40.9	58.5	32.1	38.9	56.8	24.5	28.4	43.5	28.0	26.8	42.6
SEC>T+bloom	max	27.9	49.6	64.7	35.3	39.5	57.3	<b>12.7</b>	63.3	74.3	26.2	58.9	71.6
SEC>T+bloom	emd	29.7	52.8	67.5	35.3	44.8	61.6	16.4	65.8	77.3	26.0	65.5	76.7
SEC>T+bloom	bemd	26.8	56.6	<b>70.1</b>	31.7	47.8	63.7	14.4	<b>71.6</b>	80.9	16.8	70.8	80.1
SEC>T+bloom+rank*	bemd	26.8	<b>56.7</b>	68.8	33.1	44.0	58.5	15.7	71.1	79.1	18.0	66.8	76.1
SEC>T+emb*	bemd	26.3	55.8	69.4	<b>27.5</b>	<b>48.9</b>	<b>65.1</b>	15.5	71.6	<b>81.0</b>	<b>16.2</b>	<b>71.0</b>	<b>81.1</b>

TABLE 4.4: Experimental results for topic segmentation and single-label classification on four WIKISECTION datasets.  $n = 718 / 464 / 3,907 / 2,507$  documents. Numbers are given as  $P_k$  on sentence level, micro-averaged  $F_1$  and MAP at segment-level. For methods without segmentation, we used newlines as segment boundaries (NL) and merged sections of same classes after prediction. Models marked with \* are based on pre-trained distributed embeddings.

WikiSection-headings multi-label classification		en_disease 179 topics			de_disease 115 topics			en_city 603 topics			de_city 318 topics		
model configuration	segm.	$P_k$	P@1	MAP	$P_k$	P@1	MAP	$P_k$	P@1	MAP	$P_k$	P@1	MAP
CNN>H*	max	40.9	36.7	31.5	41.3	14.1	21.1	36.9	43.3	46.7	42.2	40.9	46.5
SEC>H+bloom	bemd	35.4	35.8	38.2	36.9	31.7	37.8	20.0	65.2	62.0	23.4	49.8	53.4
SEC>H+bloom+rank	bemd	40.2	47.8	49.0	42.8	28.4	33.2	41.9	66.8	59.0	34.9	59.6	54.6
SEC>H+emb*	bemd	30.7	<b>50.5</b>	<b>57.3</b>	<b>32.9</b>	26.6	<b>36.7</b>	17.9	<b>72.3</b>	<b>71.1</b>	19.3	68.4	<b>70.2</b>
SEC>H+emb+rank*	bemd	<b>30.5</b>	47.6	48.9	42.9	<b>32.0</b>	36.4	<b>16.1</b>	65.8	59.0	<b>18.3</b>	<b>69.2</b>	58.9
SEC>H+emb@fullwiki*	bemd	42.4	9.7	17.9	42.7	(0.0)	(0.0)	20.3	59.4	50.4	38.5	(0.0)	(0.1)

TABLE 4.5: Experimental results for segmentation and multi-label classification trained with raw Wikipedia headings. Here, the task is to segment the document and predict multi-word topics from a large ambiguous target vocabulary.

and ranking loss (+rank). We have chosen network hyperparameters using grid search on the en\_disease validation set and keep them fixed over all evaluation runs. For all configurations, we set BLSTM layer size to 256, topic embeddings dimension to 128. Models are trained on the complete train splits with a batch size of 16 documents (reduced to 8 for bag-of-words), 0.01 learning rate, 0.5 dropout and ADAM optimization. We used early stopping after 10 epochs

without MAP improvement on the validation data sets. We pre-trained word embeddings with 256 dimensions for the specific tasks using word2vec on lowercase English and German Wikipedia documents using a window size of 7. All tests are implemented in Deeplearning4j and run on a Tesla P100 GPU with 16GB memory. Training a SEC+bloom model on en\_city takes roughly 5 hours, inference on CPU takes on average 0.36 seconds per document. In addition, we trained a SEC>H@fullwiki model with raw headings from a complete English Wikipedia dump<sup>10</sup>, and use this model for cross-dataset evaluation.

**Quality metrics.** We measure *text segmentation* at sentence level using the probabilistic  $P_k$  error score [Beeferman et al., 1999] which calculates the probability of a false boundary in a window of size  $k$ , lower numbers mean better segmentation. As relevant section boundaries we consider all section breaks where the topic label changes. We set  $k$  to half of the average segment length. We measure *classification performance* on section level by comparing the topic labels of all ground truth sections with predicted sections. We select the pairs by matching their positions with maximum boundary overlap. We report *micro-averaged  $F_1$*  score for single-label or *Precision@1* for multi-label classification. We further report *Mean Average Precision (MAP)*, which measures the average fraction of true labels ranked above a particular label [Tsoumakas et al., 2009].

#### 4.4.2 Experimental Results

Table 4.4 shows the evaluation results of the WIKISECTION-topics single-label classification task, Table 4.5 contains the corresponding numbers for multi-label classification. Table 4.6 shows results for topic segmentation across different datasets.

**SECTOR outperforms existing classifiers.** With our given segmentation baseline (NL), the best sentence classification model CNN achieves 52.1%  $F_1$  averaged over all datasets. SECTOR improves this score significantly by 12.4 points. Furthermore, in the setting with plain text input, SECTOR improves the CNN score by 18.8 points using identical baseline segmentation. Our model finally reaches an average of 61.8%  $F_1$  on the classification task using sentence embeddings and bidirectional segmentation. This is a total improvement of 27.8 points over the CNN model.

**Topic embeddings improve segmentation.** SECTOR outperforms C99 and TopicTiling significantly by 16.4 respectively 18.8 points  $P_k$  on average. Compared to the maximum label baseline, our model gains 3.1 points by using the bidirectional embedding deviation and 1.0 points using sentence embeddings. Overall, SECTOR misses only 4.2 points  $P_k$  and 2.6 points  $F_1$  compared to the experiments with prior newline segmentation. The third experiments reveals that our segmentation method in isolation almost reaches state-of-the-art on existing datasets and beats the unsupervised baselines, but lacks performance on cross-dataset evaluation.

<sup>10</sup>excluding all documents contained in the test sets

Segmentation	Wiki-50		Cities		Elements		Clinical
	$P_k$	MAP	$P_k$	MAP	$P_k$	MAP	$P_k$
and multi-label classification							
GraphSeg	63.6	n/a	40.0	n/a	49.1	n/a	–
BayesSeg	49.2	n/a	36.2	n/a	<b>35.6</b>	n/a	57.8
TextSeg	<b>18.2*</b>	n/a	<b>19.7*</b>	n/a	41.6	n/a	<b>30.8</b>
SEC>H+emb@en_disease	–	–	–	–	43.3	9.5	36.5
SEC>C+emb@en_disease	–	–	–	–	45.1	n/a	35.6
SEC>H+emb@en_city	30.0	31.4	28.2	<b>56.5</b>	41.0	7.9	–
SEC>C+emb@en_city	31.3	n/a	22.9	n/a	48.8	n/a	–
SEC>H+emb@cities	33.3	15.3	21.4*	52.3*	39.2	12.1	37.7
SEC>H+emb@fullwiki	28.6*	<b>32.6*</b>	33.4	40.5	42.8	<b>14.4</b>	36.9

TABLE 4.6: Experimental results for cross-dataset topic segmentation. Numbers marked with \* are generated by models trained specifically for this dataset. A value of ‘n/a’ indicates that a model is not applicable to this problem.

**Bloom filters on par with word embeddings.** Bloom filter encoding achieves high scores among all datasets and outperforms our bag-of-words baseline, possibly because of larger training batch sizes and reduced model parameters. Surprisingly, word embeddings did not improve the model significantly. On average, German models gained 0.7 points  $F_1$  while English models declined by 0.4 points compared to Bloom filters. However, model training and inference using pre-trained embeddings is faster by an average factor of 3.2.

**Topic embeddings perform well on noisy data.** In the multi-label setting with unprocessed Wikipedia headings, classification precision of SECTOR reaches up to 72.3% P@1 for 603 labels. This score is in average 9.5 points lower compared to the models trained on the small number of 25–30 normalized labels. Furthermore, segmentation performance is only missing 3.8 points  $P_k$  compared to the topics task. Ranking loss could not improve our models significantly, but achieved better segmentation scores on the headings task. Finally, the cross-domain English *fullwiki* model performs only on baseline level for segmentation, but still achieves better classification performance than CNN on the English cities dataset.

## 4.5 Discussion and Insights

Figure 4.5 shows classification and segmentation of SECTOR compared to the PV baseline.

**SECTOR captures latent topics from context.** We clearly see from NL predictions (left side of Figure 4.5) that SECTOR produces coherent results with sentence granularity, with topics emerging and disappearing over the course of a document. In contrast, PV predictions are scattered across the document. Both models successfully classify first (symptoms) and last sections (epidemiology). However, only SECTOR can capture diagnosis, prevention and treatment. Furthermore, we observe additional screening predictions in the center of the document. This section is actually labeled “Prevention | Screening” in the source document, which explains this

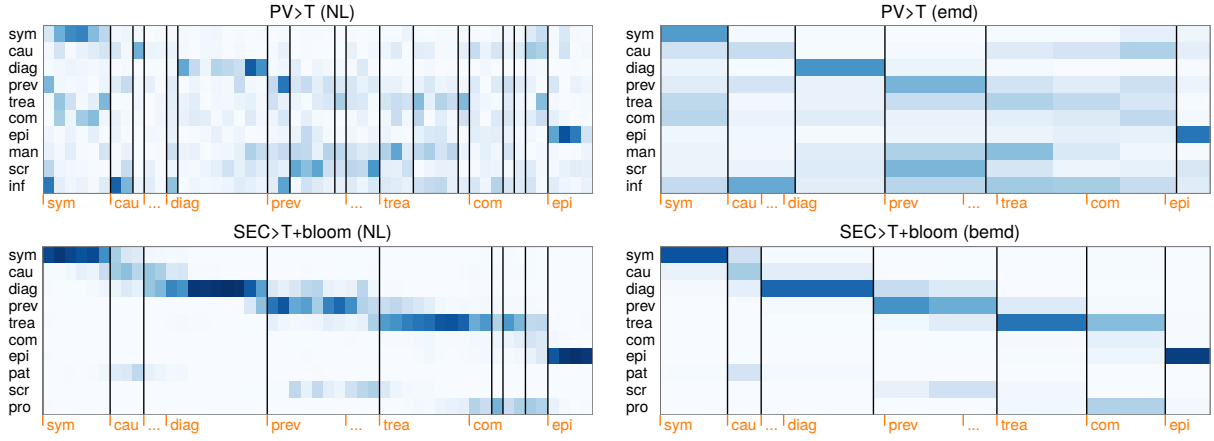


FIGURE 4.5: Heatmaps of predicted topic labels  $\hat{y}_k$  for document *Trichomoniasis* from PV and SECTOR models with newline and embedding segmentation. Shading denotes probability for 10 out of 27 selected topic classes on Y axis, with sentences from left to right. Segmentation is shown as black lines, X axis shows expected gold labels. Note that segments with same class assignments are merged in both predictions and gold standard ('...').

overlap. Furthermore, we observe low confidence in the second section labeled cause. Our multi-class model predicts for this section {diagnosis, cause, genetics}. The ground truth heading for this section is “Causes | Genetic sequence”, but even for a human reader this assignment is not clear. This shows that the multi-label approach fills an important gap and can even serve as an indicator for low-quality article structure.

Finally, both models fail to segment the complication section near the end, because it consists of an enumeration. The embedding deviation segmentation strategy (right side of Figure 4.5) completely solves this issue for both models. Our SECTOR model is giving nearly perfect segmentation using the bidirectional strategy, it only misses the discussed part of cause and is off by one sentence for the start of prevention. Furthermore, averaging over sentence-level predictions reveals clearly distinguishable section class labels.

**SECTOR represents topics coherently in vector space.** Figure 4.6 reveals an insight into the topic embedding of our SECTOR multi-class model. It is clearly visible that the model is able to separate the classes in vector space. Furthermore, the sequential classification of sentences from a single document (here depicted as line) is continuously moving through the space of sections. This makes SECTOR embeddings an ideal source for semantic language understanding tasks, because they can provide a topical discourse vector that is generated from the sequential context of an entire long document.

## 4.6 Related Work

The analysis of emerging topics over the course of a document is related to a large number of research areas. In particular, topic modeling [Blei et al., 2003] and topic detection and tracking

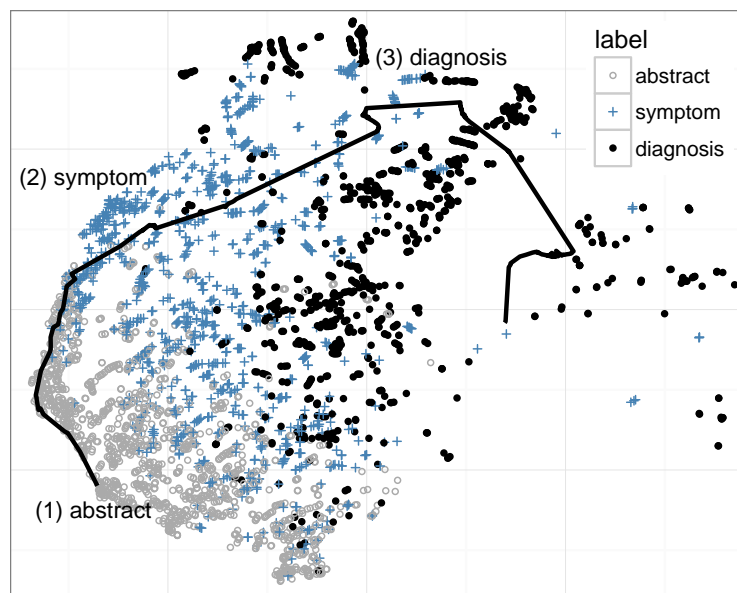


FIGURE 4.6: TSNE vector space of the headings in our SECTOR topic embedding. The path illustrates the sequence of section labels abstract, symptom, diagnosis for the document *Dyslexia*. Every dot is the projection of a randomly chosen sentence from the *en\_disease* validation set. Shape and color of the dots relate to the mentioned section labels, others are left out for readability.

(TDT) [Jin et al., 1999] focus on representing and extracting the semantic topical content of text. Text segmentation [Beeferman et al., 1999] is used to split documents into smaller coherent chunks. Finally, text classification [Joachims, 1998] is often applied to detect topics on text chunks. Our method unifies those strongly interwoven tasks and is the first to evaluate a combined topic segmentation and classification task using a dataset with long documents.

**Topic modeling** is commonly applied to entire documents using probabilistic models, such as latent Dirichlet allocation (LDA) [Blei et al., 2003]. AlSumait et al. [2008] introduced an online topic model that captures emerging topics when new documents appear. Gabrilovich and Markovitch [2007] proposed the Explicit Semantic Analysis method in which concepts from Wikipedia articles are indexed and assigned to documents. Later, and to overcome the vocabulary mismatch problem, Cimiano et al. [2009] introduced a method for assigning latent concepts to documents. More recently, Liu et al. [2016] represented documents with vectors of closely related domain keyphrases. Yeh et al. [2016] proposed a conceptual dynamic LDA model for tracking topics in conversations. Bhatia et al. [2016] utilized Wikipedia document titles to learn neural topic embeddings and assign document labels. Dieng et al. [2017] focused on the issue of long-range dependencies and proposed a latent topic model based on recurrent neural networks (RNNs). However, the authors did not apply the RNN to predict local topics.

**Text segmentation** has been approached with a wide variety of methods. Early unsupervised methods utilized lexical overlap statistics [Hearst, 1997; Choi, 2000], dynamic programming



[Utiyama and Isahara, 2001], Bayesian models [Eisenstein and Barzilay, 2008] or point-wise boundary sampling [Du et al., 2013] on raw terms. Later, supervised methods included topic models [Riedl and Biemann, 2012] by calculating a coherence score using dense topic vectors obtained by LDA. Bayomi et al. [2015] exploited ontologies to measure semantic similarity between text blocks. Alemi and Ginsparg [2015] and Naili et al. [2017] studied how word embeddings can improve classical segmentation approaches. Glavaš et al. [2016] utilized semantic relatedness of word embeddings by identifying cliques in a graph.

More recently, Sehikh et al. [2017] utilized Long Short-Term Memory (LSTM) networks and showed that cohesion between bidirectional layers can be leveraged to predict topic changes. In contrast to our method, the authors focused on segmenting speech recognition transcripts on word level without explicit topic labels. The network was trained with supervised pairs of contrary examples and was mainly evaluated on artificially-segmented documents. Our approach extends this idea so it can be applied to dense topic embeddings which are learned from raw section headings. Wang et al. [2017a] tackled segmentation by training a CNN to learn coherence scores for text pairs. Similar to Sehikh et al. [2017], the network was trained with short contrary examples and no topic objective. The authors showed that their point-wise ranking model performs well on datasets by Jeong and Titov [2010]. In contrast to our method, the ranking algorithm strictly requires a given ground truth number of segments and no topic labels are predicted.

Koshorek et al. [2018] presented a large new dataset for text segmentation based on Wikipedia that includes section headings. The authors introduced a neural architecture for segmentation which is based on sentence embeddings and four layers of BLSTM. Similar to Sehikh et al. [2017], the authors used a binary segmentation objective on sentence level, but trained on entire documents. Our work takes up this idea of end-to-end training and enriches the neural model with a layer of latent topic embeddings that can be utilized for topic classification.

**Text classification** is mostly applied at paragraph or sentence level using machine learning methods such as Support Vector Machines [Joachims, 1998] or, more recently, shallow and deep neural networks [Hoa T. Le et al., 2018; Conneau et al., 2017]. Notably, Paragraph Vectors [Le and Mikolov, 2014] is an extension of word2vec for learning fixed-length distributed representations from texts of arbitrary length. The resulting model can be utilized for classification by providing paragraph labels during training. Furthermore, Kim [2014] has shown that convolutional neural networks (CNNs) combined with pre-trained task-specific word embeddings achieve highest scores for various text classification tasks.

**Combined approaches** of topic segmentation and classification are rare to find. Agarwal and Yu [2009] approached to classify sections of BioMed Central articles into four structural classes (introduction, methods, results and discussion). However, their manually-labeled dataset only contains a sample of sentences from the documents, so they evaluated sentence classification as an isolated task. Chen et al. [2009] introduced two Wikipedia-based datasets for segmentation,

one about large cities, the second about chemical elements. While these datasets have been used to evaluate word-level and sentence-level segmentation [Koshorek et al., 2018], we are not aware of any topic classification approach on this dataset.

Tepper et al. [2012] approached segmentation and classification in a clinical domain as supervised sequence labeling problem. The documents were segmented using a Maximum Entropy model and then classified into 11 or 33 categories. A similar approach by Ajjour et al. [2017] used sequence labeling with a small number of 3–6 classes. Their model is extractive, so it does not produce a continuous segmentation over the entire document. Finally, Piccardi et al. [2018] did not approach segmentation, but recommended an ordered set of section labels based on Wikipedia articles.

Eventually, we are inspired by *passage retrieval* [Liu and Croft, 2002] as an important downstream task for topic segmentation and classification. For example, Hewlett et al. [2016] proposed WikiReading, a QA task to retrieve values from sections of long documents. The objective of TREC Complex Answer Retrieval is to retrieve a ranking of relevant passages for a given outline of hierarchical sections [Nanni et al., 2017]. Both tasks highly depend on a building block for local topic embeddings such as our proposed model.

## 4.7 Conclusions

In this chapter, we have approached RQ 2, the detection of topics and structure in long documents. We presented SECTOR, a novel neural model for coherent text segmentation and classification based on latent topics. This model supports our vision of Neural Machine Reading by extending the scope of language understanding towards higher-level abstractions on passage and document level. We further contributed WIKISECTION, a collection of four large datasets from clinical and geopolitical domains in English and German for this task. Our end-to-end method builds upon a neural topic embedding which is trained using Wikipedia headings to optimize a bidirectional LSTM classifier. We showed that our best performing model is based on sparse word features with Bloom filter encoding and significantly improves classification precision for 25–30 topics on comprehensive documents by up to 29.5 points  $F_1$  compared to state-of-the-art sentence classifiers. We used the bidirectional deviation in our topic embedding to segment a document into coherent sections without additional training. Finally, our experiments showed that extending the task to multi-label classification of over 600 ambiguous topic words still produces coherent results with 71.1% average precision. In the next chapter, we will extend this idea towards general Machine Reading by encoding named entities and aspects alongside the document structure.

## Chapter 5

# Contextualized Document Representations for Answer Retrieval

In the preceding chapters we have approached the detection of named entities (RQ 1) and topical structure (RQ 2) in domain-specific text. In this chapter, we aim to combine these two approaches towards a general document representation for Machine Reading. As a central idea, we approach RQ 3: *How can we embed discourse structure into document representations?* To examine the impact of discourse-aware representations, we will further investigate RQ 4: *How effective are document representations for retrieving answer passages?*

We present Contextual Discourse Vectors (CDV), a distributed document representation for efficient answer retrieval from long documents<sup>1</sup>. Our approach is based on structured query tuples of entities and aspects from free text and domain-specific taxonomies. Our model leverages a dual encoder architecture with hierarchical LSTM layers and multi-task training to encode the position of entities and aspects alongside the document discourse. We use our continuous representations to resolve queries with short latency using approximate nearest neighbor search on sentence level. We apply the CDV model for retrieving coherent answer passages from nine English public healthcare resources from the Web, addressing both patients and medical professionals. Because there is no end-to-end training data available for such an application scenario, we train our model with self-supervised data from Wikipedia. We compare our general model with several state-of-the-art baselines for passage ranking and discuss the adaptation to heterogeneous domains without additional fine-tuning.

This chapter is structured as follows: In Section 5.1, we introduce our idea of discourse-aware representations for Answer Passage Retrieval. In Section 5.2, we model a structured query for information-seeking tasks based on vector space representations of entities and aspects. In Section 5.3, we present the architecture of our CDV model and the process for self-supervised training. In Section 5.4, we evaluate our model in an answer retrieval task over nine English healthcare text resources. In Section 5.5, we discuss the results and provide an analysis of the model’s false predictions. In Section 5.6, we summarize related work. We conclude in Section 5.7 with a reflection on our research questions for Neural Machine Reading.

---

<sup>1</sup>This chapter was published and presented by S. Arnold, B. van Aken, P. Grundmann, F. A. Gers, and A. Löser [2020]. “Learning Contextualized Document Representations for Healthcare Answer Retrieval”. In: *Proceedings of The Web Conference 2020*. ACM, pp. 1332–1343

## 5.1 Introduction

In a clinical decision support system (CDSS), doctors and healthcare professionals require access to information from heterogeneous sources, such as research papers [Gorman et al., 1994; Schardt et al., 2007], electronic health records [Hanauer et al., 2015], clinical case reports [Fujiwara et al., 2018], reference works and knowledge base articles. Differential diagnosis is an important task where a doctor seeks to retrieve answers for non-factoid queries about diseases, such as “symptoms of IgA nephropathy” (see Figure 5.1). A relevant answer typically spans multiple sentences and is most likely embedded into the discourse of a long document [Yang et al., 2016b; Cohan et al., 2018].

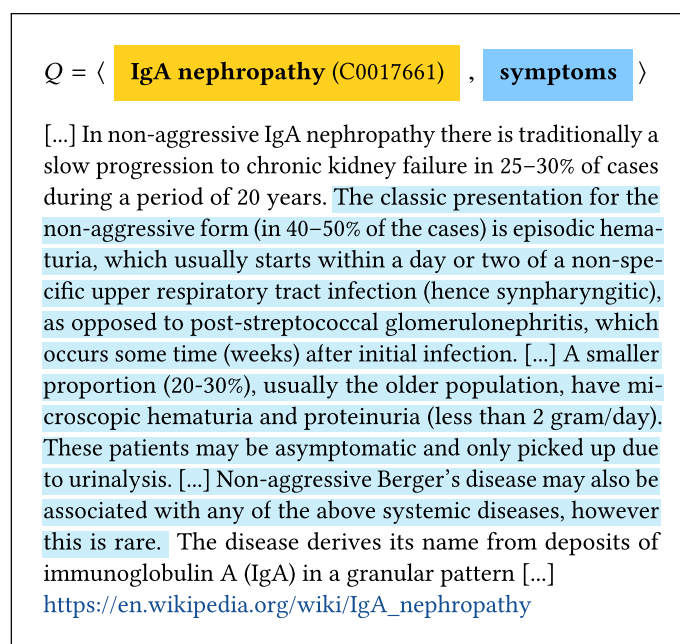


FIGURE 5.1: Example of a structured entity/aspect query  $Q$  and a highlighted answer passage from Wikipedia. Note that the answer is part of a longer document and there is almost no word overlap between query and answer passage.

Evidence-based medicine (EBM) has made efforts to structure physicians’ information needs into short structured question representations, such as PICO (patient, intervention, comparison, outcome) [Richardson et al., 1995] and—more general—well-formed background–foreground questions [Cheng, 2004]. We support this important query intention and define a query as structured tuple of *entity* (e.g. a disease or health problem) and *aspect*. Our model is focused on clinical aspects such as therapy, diagnosis, etiology, prognosis, and others, which have been described in the literature previously by manual clustering of semantic question types [Huang et al., 2006] or crawling medical Wikipedia section headings [Arnold et al., 2019]. In a CDSS, a doctor can express these query terms with identifiers from a knowledge base or medical taxonomy, e.g. UMLS, ICD-10 or Wikidata. The system will support the user in assigning these links by search and auto-completion operators [Schneider et al., 2018; Fujiwara et al., 2018], which allows us to use these representations as input for the answer retrieval task.

### 5.1.1 Challenges for Answer Passage Retrieval

Several methods have been proposed to apply deep neural networks for effective information retrieval [Guo et al., 2016; Mitra et al., 2017; Dai et al., 2018] and question answering [Seo et al., 2017; Wang et al., 2017b], also with focus on healthcare [Zhu et al., 2019; Jin et al., 2019]. However, our scenario poses a unique combination of open challenges to a retrieval system:

1. *Task coverage*: Query intentions span a broad range in specificity and complexity [Huang et al., 2006; Nanni et al., 2017]. For example, medical specialists may pose very precise queries that align with a pre-defined taxonomy and focus on rare diseases. On the other hand, nursing staff might have broader and more heterogeneous questions. However, in most cases we do not have access to task-specific training data, so training the model for a single intention is not feasible. We therefore require a generalized query representation that covers a broad range of intents and taxonomies, even with limited training data.
2. *Domain adaptability*: In many cases we do not even have textual data readily available at training time from all resources in a CDSS. However, we observe linguistic and semantic shifts between the heterogeneous types of text, e.g. different use of terms and abbreviations among groups of doctors. Therefore we face a zero-shot retrieval task that requires robust domain transfer abilities across diverse biomedical, clinical and healthcare text resources [Logeswaran et al., 2019].
3. *Contextual coherence*: Answers are often expressed as passages in context of a long document. Therefore the model needs to respect long-range dependencies such as the sequence of micro-topics that establish a coherent ‘train of thought’ in a document [Arora et al., 2016; Arnold et al., 2019]. At the same time, the model is required to operate on a fine granularity (e.g., on sentence level) rather than on entire documents to be able to capture the boundaries of answers [Keikha et al., 2014].
4. *Efficient neural information retrieval*: All documents in the CDSS need to be accessible with fast ad-hoc queries by the users. Many question answering models are based on pairwise similarity, which is computationally too intensive when applied to large-scale retrieval tasks [Gillick et al., 2018]. Instead, we require a continuous retrieval model that allows for offline indexing and approximate nearest neighbor search with high recall [Gillick et al., 2018], even for rare queries and with low latency in the order of milliseconds.

### 5.1.2 Contextual Discourse Vectors

We approach these challenges and present *Contextual Discourse Vectors* (CDV)<sup>2</sup>, a neural document representation which is based on discourse modeling and fulfills the above requirements. Our method is the first to address answer retrieval with structured queries on long heterogeneous documents from the healthcare domain.

<sup>2</sup>Code and evaluation data is available at <https://github.com/sebastianarnold/cdv>

CDV is based on hierarchical layers to encode word, sentence and document context with bidirectional Long Short-Term Memory (BLSTM). The model uses multi-task learning [Caruana, 1997] to align the sequence of sentences in a long document to the clinical knowledge encoded in pre-trained entity and aspect vector spaces. We use a dual encoder architecture [Gillick et al., 2018], which allows us to precompute discourse vectors for all documents and later answer ad-hoc queries over that corpus with short latency [Gillick et al., 2019]. Consequently, the model predicts similarity scores with sentence granularity and does not require an extra inference step after the initial document indexing.

We apply our CDV model for retrieving passages from various public health resources on the Web, including NIH documents and Patient articles, with structured clinical query intentions of the form  $\langle \text{entity}, \text{aspect} \rangle$ . Because there is no training data available from most sources, we use a self-supervised approach to train a generalized model from medical Wikipedia texts. We apply this model to the texts in our evaluation in a zero-shot approach [Palatucci et al., 2009] without additional fine tuning. In summary, our major contributions include:

- We propose a structured entity/aspect healthcare query model to support the essential query intentions of medical professionals. Our task is focused on the efficient retrieval of answer passages from long documents of heterogeneous health resources.
- We introduce CDV, a contextualized document representation for passage retrieval. Our model leverages a dual encoder architecture with BLSTM layers and multi-task training to encode the position of discourse topics alongside the document. We use the representations to answer queries using nearest neighbor search on sentence level.
- Our model utilizes generalized language models and aligns them with clinical knowledge from medical taxonomies, e.g. pre-trained entity and aspect embeddings. Therefore, it can be trained with sparse self-supervised training data, e.g. from Wikipedia texts, and is applicable to a broad range of texts.
- We prove the applicability of our CDV model with extensive experiments and a qualitative error analysis on nine heterogeneous healthcare resources. We provide additional entity/aspect labels for all datasets. Our model significantly outperforms existing document matching methods in the retrieval task and can adapt to different healthcare domains without fine-tuning.

## 5.2 Discourse Model

Our first challenge is to design a query model which can adapt to a broad number of healthcare answer retrieval tasks and utilizes the information sources available in a CDSS. In this section, we introduce a vector-space representation for this purpose.

We define a query as a structured tuple  $Q = \langle \text{entity}, \text{aspect} \rangle$ . This approach of using two complementary query arguments originates from the idea of structured background–foreground questions in EBM [Cheng, 2004] and has been used before in many triple-based retrieval systems [Adolphs et al., 2011]. In our healthcare scenario, we restrict entities to be of type *disease*, e.g. IgA nephropathy, and aspects from the clinical domain, e.g. symptoms, treatment, or prognosis. We discuss these two spaces and their combination in this section. In general, our model is not limited to the query spaces used in this paper and further extendable to a larger number of arguments.

### 5.2.1 Entity Representation

The first part of our problem is to represent the entity in focus of a query. In contrast to interaction-based models, which are applied to query–document pairs, our approach is to decouple entity encoding and document encoding. Therefore we follow recent work in representation-based Entity Linking [Gillick et al., 2019] and embed textual knowledge from the clinical domain into this representation. Our goal is to generalize entity representations, so the model will be able to align to existing taxonomies without retraining. Therefore, our entity space must be as complete as possible: it needs to cover each of the entities that appear in the discourse training data, but also rare entities that we expect at query time, e.g. in the application. We must further provide a robust method for predicting unseen entities [Logeswaran et al., 2019]. In contrast to highly specialized entity embeddings constructed from knowledge graphs or multimodal data [Beam et al., 2018], our general approach is based on textual data and allows us to apply the model to different knowledge bases and domains.

**Entity Embeddings.** Our goal is to create a mapping of each entity in the knowledge base  $E \in \mathcal{K}$  identified by its ID into a low-dimensional entity vector space  $\mathbb{E} \subset \mathbb{R}^d$ . We train an embedding by minimizing the loss for predicting the entity from sentences  $s \in \mathcal{D}_E$  in the entity descriptions:

$$\mathcal{L}_{\text{entity}}(\Theta) = -\log \prod_{E \in \mathcal{K}} \prod_{s \in \mathcal{D}_E} p_{\Theta}(E.\text{id} \mid s) \quad (5.1)$$

where  $\Theta$  denotes the parameters required to approximate the probability  $p$ . We optimize  $\Theta$  using a bidirectional Long Short-Term Memory (BLSTM) [Hochreiter and Schmidhuber, 1997] to predict the entity ID  $E.\text{id}$  from the words  $w_i \in s$ . We encode  $w_i$  using Fasttext embeddings [Bojanowski et al., 2017] and use Bloom filters [Serrà and Karatzoglou, 2017] to compress  $E.\text{id}$  into a hashed bit encoding, allowing for less model parameters and faster training.

$$\begin{aligned} p_{\Theta}(E.\text{id} \mid s) &= p_{\Theta}(E.\text{id} \mid w_1, \dots, w_N) \\ &\approx p(\text{bloom}(E.\text{id}) \mid \text{BLSTM}_{\Theta}(w_1, \dots, w_N)) \end{aligned} \quad (5.2)$$

---

<sup>3</sup>we use  $d$  as a placeholder for all embedding vector sizes, even if they are not equal

Subsequently, we extend the approach of Palangi et al. [2016] and define the embedding function  $\epsilon$  as the average output of the hidden word states  $\vec{g}_k$  and  $\tilde{g}_k$  at the first respectively last time step:

$$\vec{g}_k = \text{LSTM}_\Theta(\vec{g}_{k-1}, w_k) \quad (5.3)$$

$$\tilde{g}_k = \text{LSTM}_\Theta(\tilde{g}_{k+1}, w_k)$$

$$\epsilon(s) = \frac{\vec{g}_T + \tilde{g}_1}{2} \quad (5.4)$$

Finally, we generate *entity embeddings*  $\epsilon_E \in \mathbb{E}$  by applying the embedding function to all descriptions available. In case of unseen entities, the embedding can be generated on-the-fly:

$$\epsilon_E = \begin{cases} \frac{1}{|\mathcal{D}_E|} \sum_{s \in \mathcal{D}_E} \epsilon(s) & \text{if } E \in \mathcal{K} \\ \epsilon(E.\text{mention}) & \text{if } E \notin \mathcal{K} \end{cases} \quad (5.5)$$

**Training data for clinical named entities.** We train the entity representation for diseases, syndromes and health problems using textual descriptions from various sources: Wikidata<sup>4</sup>, UMLS<sup>5</sup>, GARD<sup>6</sup>, Wikipedia abstracts, and the Diseases Database<sup>7</sup>. In total, the knowledge base contains over 27,000 entities identified by their Wikidata ID. We trained roughly 9,700 common entities with text from Wikipedia abstracts, while we used for rare entities only their name and short description texts.

### 5.2.2 Aspect Representation

The second part of our problem is to represent the aspect in the query tuple. Here, we expect a wide range of clinical facets and we do not want to limit the users of our system to a specific terminology. Instead, we train a low-dimensional aspect vector space  $\mathbb{A} \subset \mathbb{R}^d$  using the Fasttext skip-gram model [Bojanowski et al., 2017] on medical Wikipedia articles. This approach places words with similar semantics nearby in vector space and allows queries with morphologic variations using subword representations.

**Aspect embeddings.** To find all possible aspects, we adopt prior work [Arnold et al., 2019] and collect all section headings from the medical Wikipedia articles. These headings typically consist of 1–3 words and describe the main topic of a section. We apply moderate preprocessing (lowercase, remove punctuation, split at “and|&”) to generate *aspect embeddings*  $\alpha_A \in \mathbb{A}$  using a BLSTM encoder  $\alpha(s)$  with the same architecture as discussed above:

$$\alpha_A = \frac{1}{|\mathcal{D}_A|} \sum_{s \in \mathcal{D}_A} \alpha(s) \quad (5.6)$$

<sup>4</sup><https://www.wikidata.org/wiki/Q12136>

<sup>5</sup><https://uts.nlm.nih.gov>

<sup>6</sup><https://rarediseases.info.nih.gov>

<sup>7</sup><http://www.diseasesdatabase.com>



Heading	cusum	Heading	cusum
information (abstract)	9.6%	mechanism	54.1%
treatment	16.0%	culture	54.7%
diagnosis	22.3%	society	55.3%
symptoms	28.2%	research	55.9%
signs	33.0%	risk factors	56.4%
causes	37.1%	presentation	56.9%
history	39.3%	differential diagnosis	57.3%
pathophysiology	41.4%	surgery	57.7%
management	43.4%	treatments	58.1%
epidemiology	45.4%	pathogenesis	58.4%
cause	47.3%	medications	58.7%
classification	48.9%	complications	59.0%
prognosis	50.4%	characteristics	59.3%
prevention	51.7%	medication	59.5%
types	52.5%	other animals	59.7%
genetics	53.3%	pathology	60.0%

TABLE 5.1: Distribution of the top 32 headings (60% of 577K total occurrences) contained in our training set. Numbers are given as cumulative sum. We observe that these headings cover the most important aspects for differential diagnosis.

**Training data for clinical aspects.** We train the embedding with over 577K sentences from Wikipedia (see Table 5.1). We observe that there is a vocabulary mismatch in the headings so that potentially synonymous aspects are frequently labeled with different headings, e.g. types / classification or signs / symptoms / presentation / characteristics. However, it is also possible that in some contexts these aspects are hierarchically structured, e.g. presentation refers to the visible forms of a symptom. Our vector-space representation reflects these similarities, so it is possible to distinguish between these nuances at query time.

### 5.2.3 Query Representation

Finally, we represent the query as a tuple of entity and aspect embeddings using vector concatenation ( $\oplus$ ):

$$Q(E, A) = \langle \epsilon_Q \in \mathbb{E}, \alpha_Q \in \mathbb{A} \rangle = \epsilon_Q \oplus \alpha_Q \quad (5.7)$$

This query encoder constitutes the upper part of our dual encoder architecture shown in Figure 5.2. Our next goal is to find the positions in all documents where the local discourse matches the query  $Q$ . In the next section, we introduce our contextualized document representation that allows similarity matching between  $Q$  and each document at sentence level.

## 5.3 Contextualized Document Representation

In this section, we introduce Contextual Discourse Vectors (CDV), a distributed document representation that focuses on coherent encoding of local discourse in the context of the entire document. The architecture of our model is shown in Figure 5.2. We approach the challenges

introduced in Section 5.1 by reading a document at word and sentence level (Section 5.3.1) and encoding sentence-wise representations using recurrent layers at document level (Section 5.3.2). We use the representations to measure similarity between every position in the document and the query (Section 5.3.3). Our model is trained to match the entity/aspect vector spaces introduced in Section 5.2 using self-supervision (Section 5.3.4).

### 5.3.1 Sentence Encoder

The first group of layers in our architecture encodes the plain text of an entire document into a sequence of vector representations. As we expect long documents—the average document length in our test sets is over 1,200 words—we chose to reduce the computational complexity by encoding the document discourse at sentence-level. It is important to avoid losing document context and word–discourse interactions (e.g. entity names or certain aspect-specific terms) during this step. Furthermore, our challenge of *domain adaptability* requires the sentence encoder to be robust to linguistic and semantic shifts from text sources that differ from the training data.

Therefore we start at the input layer by encoding all words in a document  $D$  into fixed low-dimensional word vectors  $w_{1...N} \in \mathbb{R}^d$  using pre-trained word embeddings with subword information (see below). Next, we encode all sentences  $s_{1...T} \in D$  into sentence representations  $\sigma_t \in \mathbb{R}^d$  based on the words  $w_k \in s_t$  in the sentence. This will reduce the number of computational time steps from  $N$  words in a document to  $T$  sentences. We compare two approaches for this sentence encoding step:

**Compositional sentence embeddings.** As the simplest approach we use an average vector composition of the word embeddings  $w_k$  from GloVe [Pennington et al., 2014] or Fasttext [Bojanowski et al., 2017], which is more robust against out-of-vocabulary errors:

$$\sigma_{\text{avg}}(s) = \frac{1}{\text{len}(s)} \sum_{w_k \in s} w_k \quad (5.8)$$

**Pooling-based sentence embeddings.** Since we want the model to be able to focus on individual words, we apply a language model encoder. We use the recent BioBERT [Lee et al., 2019], a transformer model which is pre-trained with a large amount of biomedical context on sub-word level. To generate sentence vectors from the input sequence, we use pooling of the attention layers per sentence:

$$\sigma_{\text{pool}}(s) = \text{BioBERT}(w_{s.\text{begin}}, \dots, w_{s.\text{end}}) \quad (5.9)$$

Finally, we concatenate a positional encoding to the sentence embeddings, which encodes some rule-based structural flags such as begin/end-of-document, begin/end-of-paragraph, is-list-item. This encoding helps to guide the document encoder through the structure of a document.

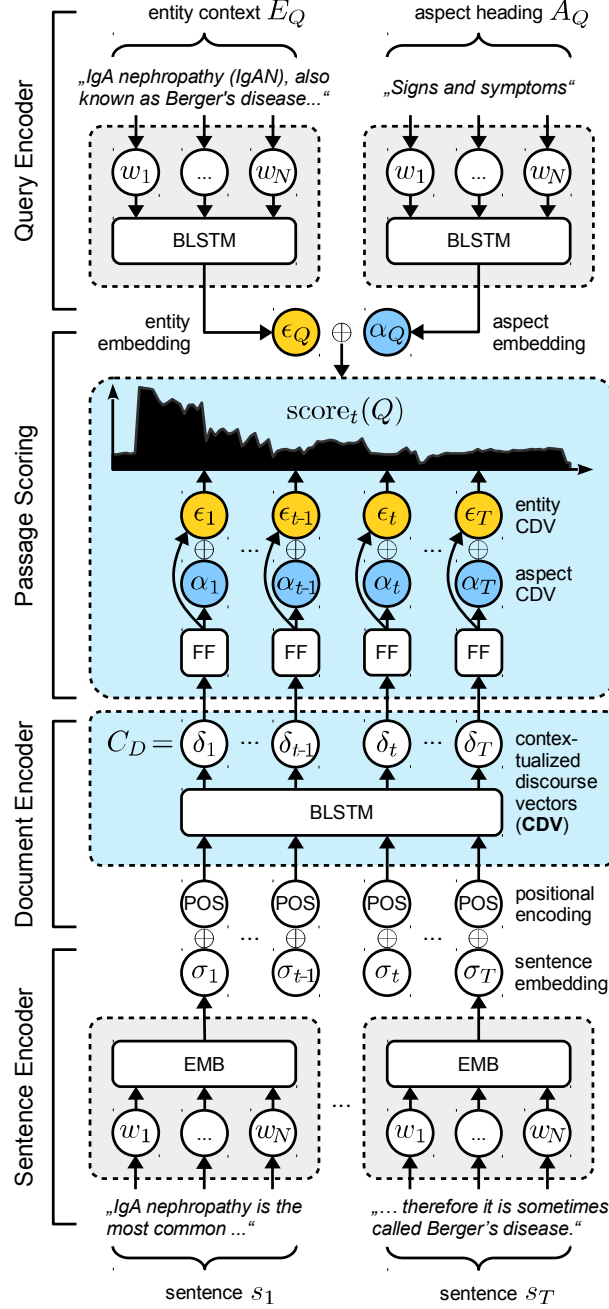


FIGURE 5.2: Neural network architecture for our contextualized document representation. The contextual discourse vectors (CDV) are generated by a hierarchical stack of layers: sentence encoder (GloVe/Fasttext/BioBERT) and document encoder (BLSTM). The query encoder (entity/aspect embeddings) is used for scoring on sentence level.

### 5.3.2 Document Encoder

The second group of layers in our architecture encodes the sequence of sentences over the document. The objective of these layers is to transform the word/sentence input space into discourse vector space—which will later match with query entity and aspect spaces—in the context of the document. To achieve *contextual coherence*, we use the entire document as input for a recurrent neural network with parameters  $\Theta$ , which we optimize at training time to minimize the loss over the sequence:

$$\mathcal{L}_{\text{doc}}(\Theta) = -\log \prod_{t=1}^T p_{\Theta}(\epsilon(s_t), \alpha(s_t) \mid \sigma(s_1), \dots, \sigma(s_T)) \quad (5.10)$$

We adopt the architecture of SECTOR [Arnold et al., 2019] and use bidirectional LSTMs to read the document sentence-by-sentence. We use a final dense layer (matrix  $\mathbf{W}_{he}$  and bias  $\mathbf{b}_e$ ) to produce the *local discourse vectors*  $\delta_{1..T}$  for every sentence in  $D$ .

$$\begin{aligned} \vec{h}_t &= \text{LSTM}_{\Theta}(\vec{h}_{t-1}, \sigma(s_t)) \\ \tilde{h}_t &= \text{LSTM}_{\Theta}(\tilde{h}_{t+1}, \sigma(s_t)) \\ \delta_t &= \tanh(\mathbf{W}_{he}(\vec{h}_t \oplus \tilde{h}_t) + \mathbf{b}_e) \end{aligned} \quad (5.11)$$

The CDV matrix  $\mathbf{C}_D = [\delta_1, \dots, \delta_T]$  is our discourse-aware document representation which embeds all features necessary to decode contextualized entity and aspect information for  $D$ .

### 5.3.3 Passage Scoring

The center layer in our architecture addresses our main task objective: find the passages with highest similarity to the query. In Section 5.3, we described generalized vector spaces for entities  $\epsilon \in \mathbb{E}$  and aspects  $\alpha \in \mathbb{A}$  that we use as query representation  $Q$  for high *task coverage*. We train our discourse vectors  $\mathbf{C}_D$  to share the same vector spaces  $\mathbb{E}$  and  $\mathbb{A}$ . This enables us to run *efficient neural information retrieval* of multiple ad-hoc queries  $Q$  over the pre-computed CDV vectors  $\mathbf{C}_D$  later without having to re-run inference on the document encoder for each query. We store all vectors  $\delta_t$  in an in-memory vector index that allows us to efficiently retrieve approximate nearest neighbors using cosine distance. Figure 5.3 shows the overall process of training, indexing and ad-hoc answer retrieval. Because we reuse entity and aspect embeddings for training, our document model ‘inherits’ the properties from these spaces, e.g. robustness for unseen and rare entities or aspects.

**Discourse decoder.** To decode the individual entity and aspect predictions  $\hat{\epsilon}_t, \hat{\alpha}_t$  from  $\delta_t \in \mathbf{C}_D$ , we utilize two learned decoder matrices  $\mathbf{W}_{\delta\epsilon}, \mathbf{W}_{\delta\alpha}$  with bias terms  $\mathbf{b}_{\epsilon}, \mathbf{b}_{\alpha}$ . We optimize these parameters by using a multi-task objective with shared weights [Caruana, 1997] to minimize

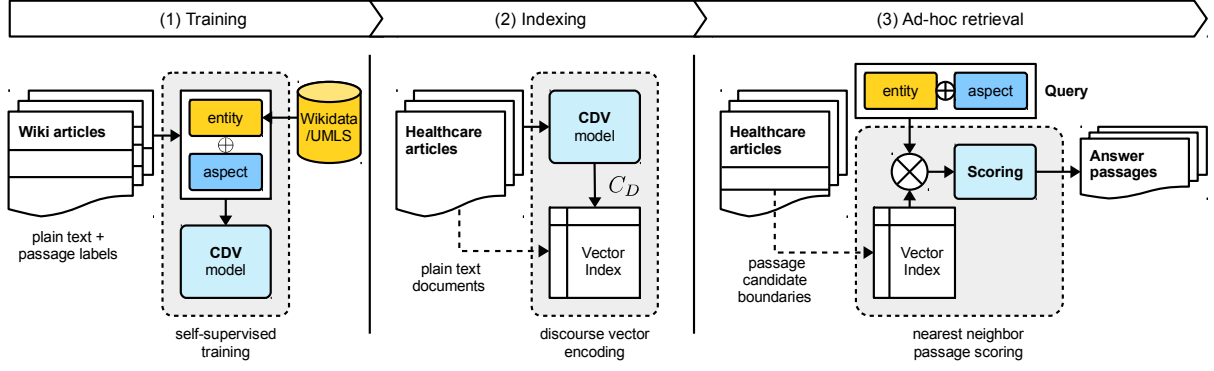


FIGURE 5.3: The entire Answer Passage Retrieval process with three stages. (1) We train the discourse-aware document representation model using self-supervision on Wikipedia articles. (2) At indexing time, the model is applied once to the entire test corpus of unseen documents. The discourse vectors are saved into a vector index. (3) A query is retrieved by ranking similarity scores between the query representation and all sentence-level vectors in the candidate passages.

the distance to the training labels  $\epsilon_t, \alpha_t$ :

$$\begin{aligned}
 \hat{\epsilon}_t &= \tanh(\mathbf{W}_{\delta\epsilon}\delta_t + \mathbf{b}_\epsilon) \\
 \hat{\alpha}_t &= \tanh(\mathbf{W}_{\delta\alpha}\delta_t + \mathbf{b}_\alpha) \\
 \mathcal{L}_{\text{cdv}}(\Theta) &= \frac{1}{T} \sum_{t=1}^T (\|\hat{\epsilon}_t - \epsilon_t\| + \|\hat{\alpha}_t - \alpha_t\|)
 \end{aligned} \tag{5.12}$$

**Sentence scoring.** To compute similarity scores at query time, we pick up our query representation (Eq. 5.7) and compute the semantic similarity between  $Q$  and each contextual discourse vector  $\delta_t$  in the vector index. To achieve low latency, we use cosine similarity between the decoded entity and aspect representations:

$$\begin{aligned}
 \text{score}_t(Q(E, A), \mathbf{C}_D) &= \text{cosine}(\epsilon_Q \oplus \alpha_Q, \hat{\epsilon}_t \oplus \hat{\alpha}_t) \\
 &= \frac{(\epsilon_Q \oplus \alpha_Q)^\top (\hat{\epsilon}_t \oplus \hat{\alpha}_t)}{\|\epsilon_Q \oplus \alpha_Q\| \|\hat{\epsilon}_t \oplus \hat{\alpha}_t\|}
 \end{aligned} \tag{5.13}$$

**Answer passage retrieval.** The scoring operation  $\text{score}_{1\dots T}(Q, \mathbf{C}_D) \in [0, 1]$  yields a sentence-level histogram which describes the similarity between query and every sentence in a document. At this point, we have the opportunity to select a coherent set of sentences as answers similar to Arnold et al. [2019]. However, because all healthcare datasets that we use for evaluation already provide passage boundaries, we leave this step for future work. Instead, we use the average sentence score per passage for answer retrieval:

$$\text{score}(Q, P) = \frac{1}{\text{len}(P)} \sum_{s_t \in P} \text{score}_t(Q, \mathbf{C}_D) \tag{5.14}$$

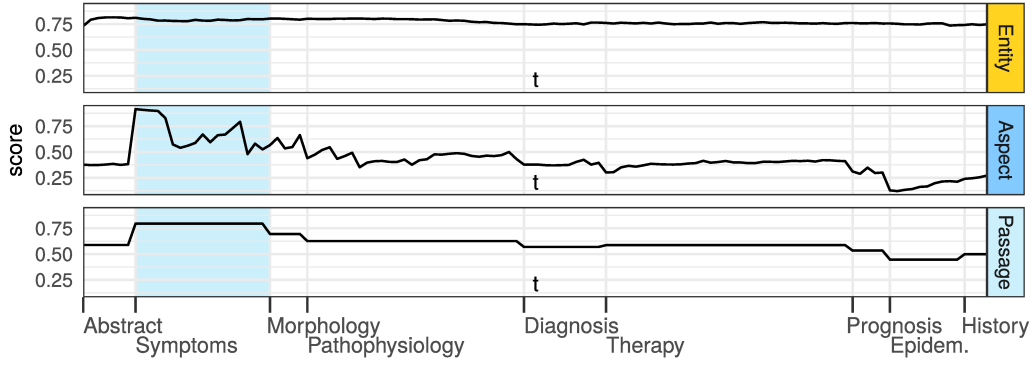


FIGURE 5.4: CDV model predictions for query “symptoms of IgA nephropathy” on the example document. The histogram shows the similarity score of the discourse vector with the query over sentences  $t = 1 \dots T$  from left to right.

Figure 5.4 shows the scoring curves divided into entity  $Q(E)$ , aspect  $Q(A)$  and an average  $\text{score}(Q(E, A), P)$ . It is clearly visible that the model coherently predicts long-range dependencies for the entity IgA nephropathy over the entire document. The aspect similarity with symptoms is much more focused on single sentences.

### 5.3.4 Self-supervised Training

We train a generalized CDV model by jointly optimizing all model parameters from sentence encoder, document encoder and passage scoring layers on a training set.

**Generating entity and aspect labels from Wikipedia.** For this task, we use the textual data about diseases and health problems available from Wikipedia. This process is self-supervised, because there exist no labeled query-answer pairs for these documents. Instead, we assign for each sentence  $s_t \in D$  a set of related entities  $E$  and aspects  $A$  using simple heuristics:

$$\begin{aligned} E(s_t, D) &= \{E \mid \text{title}(D) = E \vee \text{contains\_link}(s_t, E)\} \\ A(s_t, D) &= \{A \mid \text{heading}(s_t) = A\} \end{aligned} \quad (5.15)$$

We collected over 8,600 articles for training and removed all instances contained in any of the test sets. The collection covers over 8K entities and 15K aspects (see Table 5.2).

**Discourse decoder objective.** We create the target objectives for training using the average of the label embeddings contained in the training entities  $E(s_t, D)$  and aspects  $A(s_t, D)$ . We formulate the objectives on sentence level:

$$\begin{aligned} \epsilon_t &= \frac{\sum_{E \in E(s_t, D)} \epsilon_E}{|E(s_t, D)|} \\ \alpha_t &= \frac{\sum_{A \in A(s_t, D)} \alpha_A}{|A(s_t, D)|} \end{aligned} \quad (5.16)$$

**Optimized loss function.** We observe a strong imbalance of entity and aspect labels over the course of a single document, for example when passages contain lists (very short sentences), rare entities or have uncommon headlines. To give the network the ability to capture these anomalies, especially with larger batch sizes, we use a robust loss function [Barron, 2019] which resembles a smoothed form of Huber Loss [Huber, 1992]:

$$\mathcal{L}_{\text{cdv}+}(\Theta) = \frac{1}{T} \sum_{t=1}^T \left( \sqrt{1 + \left( \frac{\|\hat{\epsilon}_t - \epsilon_t\| + \|\hat{\alpha}_t - \alpha_t\|}{4} \right)^2} - 1 \right) \quad (5.17)$$

In the next section, we apply our CDV model to a healthcare answer retrieval task.

## 5.4 Evaluation

We evaluate our CDV model and 14 baseline methods in an Answer Passage Retrieval task. All models are trained using self-supervision on Wikipedia texts and applied as zero-shot task [Palatucci et al., 2009] (i.e. without further fine-tuning) to three diverse English healthcare datasets *WikiSection*, *MedQuAD* and *HealthQA*.

### 5.4.1 Evaluation Set-up

As queries, we use tuples of the form  $\langle \text{entity}, \text{aspect} \rangle$ . Because our task requires to retrieve the answers from over 4,000 passages and the interaction-based models in our comparison require computationally expensive pairwise inference, we evaluate all numbers on a re-ranking task [Gillick et al., 2018]. We follow the setup of Logeswaran et al. [2019] and use BM25 [Robertson et al., 1995] to provide each model with a pre-filtered set of 64 potentially relevant passage candidates<sup>8</sup>. To facilitate full recall in this model comparison, we add missing true answers to the candidates if necessary by overwriting the lowest-ranked false answers in the list and shuffle afterwards. We rank the candidate answers using exhaustive nearest neighbor search and leave the evaluation of indexing efficiency for future work. Next, we describe the datasets, metrics and methods used in our experiments.

**Evaluation datasets.** We conduct experiments on three English datasets from the clinical and healthcare domain. From the documents provided, we use the plain text of the entire document body during model inference and the segmentation information for generating the passage candidates. From all queries provided, we use the entity labels (mention text, Wikidata ID) and aspect labels (UMLS canonical name). If entity and aspect identifiers were not provided by the dataset, we added them manually by asking three annotators from clinical healthcare to label them. Table 5.2 shows an overview of the datasets.

<sup>8</sup>This choice covers 80-91% of all true answers (depending on the dataset) as trade-off between task complexity and real-world applicability. The numbers reported for HealthQA in the original paper were evaluated by re-ranking ten candidates (one relevant, 3 partially relevant and 6 irrelevant) and are therefore not comparable.

Dataset Split	Wikipedia train	⇒	WS test	MQ test	HQ test
# documents	8,605		716	1,111	178
# passages	53,477		4,373	3,762	1,109
# entities	8,605		716	1,100	221
# aspects	15,028		27	15	21
# queries	N/A		4,178	3,294	1,045
avg words/doc	977.6		1,396.7	811.1	1,449.4
avg sents/doc	43.5		63.7	48.0	82.5
avg passages/doc	6.2		6.1	3.4	6.2
avg words/passage	221.8		228.6	237.9	232.6
avg sents/passage	9.8		10.4	14.1	13.2
avg words/sent	22.7		21.9	16.9	17.6

TABLE 5.2: Statistics of our training and evaluation data sets. The training splits on the healthcare datasets are only used for model evaluation. For the final model, we only used Wikipedia for training.

*WikiSectionQA* [Arnold et al., 2019] (WS) is a large subset of full-text Wikipedia articles about diseases, labeled with entity identifiers, section headlines and 27 normalized aspect classes. We extended this dataset for answer retrieval by constructing query tuples from every section containing the given entity ID and normalized aspect label. We included abstracts as information, but skipped sections labeled as other. We use the en\_disease-test split for evaluation and made sure that none of the documents are contained in our training data.

*MedQuAD* [Abacha and Demner-Fushman, 2019] (MQ) is a collection of medical question-answer pairs from multiple trusted sources of the National Institutes of Health (NIH): National Cancer Institute (NCI)<sup>9</sup>, Genetic and Rare Diseases (GARD)<sup>10</sup>, Genetics Home Reference (GHR)<sup>11</sup>, National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK)<sup>12</sup>, National Institute of Neurological Disorders and Stroke (NINDS)<sup>13</sup>, NIH Senior Health<sup>14</sup> and National Heart, Lung and Blood Institute (NHLBI)<sup>15</sup>. We left out documents from Medline Plus due to property rights. Questions are annotated with structured identifiers for entities (UMLS CUI), aspect (semantic question type) and contain a long passage as answer. To make this dataset applicable to our method, we reconstructed the entire documents from the answer passages and kept only questions about diseases for evaluation. We filtered out documents with only one passage (these were always labeled “information”) and separated a random 25% test split from the remaining documents.

*HealthQA* [Zhu et al., 2019] (HQ) is a collection of consumer health question-answer pairs crawled from the website Patient<sup>16</sup>. The answer passages were generated from sections in

<sup>9</sup><https://www.cancer.gov>

<sup>10</sup><https://rarediseases.info.nih.gov>

<sup>11</sup><https://ghr.nlm.nih.gov>

<sup>12</sup><http://www.niddk.nih.gov/health-information/health-topics/>

<sup>13</sup><http://www.ninds.nih.gov/disorders/>

<sup>14</sup><http://nihseniorhealth.gov/>

<sup>15</sup><http://www.nhlbi.nih.gov/health/>

<sup>16</sup><https://patient.info>



the documents and annotated by human labelers with natural language questions. We reconstructed the full documents from these sections. Additionally, our annotators added structured entity and aspect labels to all questions in the test split. Although some questions are not about diseases, we kept all of them to remain comparable with related work.

**Baseline methods.** We evaluate two term-based matching functions as baseline: TF-IDF [Jones, 1972] and BM25 [Robertson et al., 1995]. We used the implementation in Apache Lucene 8.2.0<sup>17</sup> to retrieve passages containing entity and aspect of a query, e.g. “IgA nephropathy symptoms” from the index of all passages in the test dataset.

Additionally, we evaluate the following document matching methods from the literature: ARC-I and ARC-II [Hu et al., 2014], DSSM [Huang et al., 2013], C-DSSM [Shen et al., 2014], DRMM [Guo et al., 2016], MatchPyramid [Pang et al., 2016], aNMM [Yang et al., 2016a], Duet [Mitra et al., 2017], MVLSTM [Wan et al., 2016], KNRM [Xiong et al., 2017], CONV-KNRM [Dai et al., 2018] and HAR [Zhu et al., 2019]. For implementing these models, we followed Zhu et al. [2019] and used the open source implementation MatchZoo [Guo et al., 2019] with pre-trained glove.840B.300d vectors [Pennington et al., 2014]. All models were trained with our self-supervised Wikipedia training set using queries containing the entity and lowercase heading, e.g. “IgA nephropathy ; symptoms” and applied to the test sets using queries of the same structure, instead of natural language questions.

**Quality metrics.** For all ranking experiments, we use *Recall at top K* (R@K) and *Mean Average Precision* (MAP) metrics. While R@1 measures if the top-1 answer is correct or not (similar to a question answering task), we also report R@10, which corresponds with the ability to retrieve all correct answers in a top-10 results list, and MAP, which considers the entire result list.

**Implementation details.** We implement our models with the following configurations. Where applicable, we chose the hyperparameters using grid search on the WikiSection validation set:

For the sentence encoding, we use either glove.6B.300d pre-trained GloVe vectors (+avg-glove), 128d fine-tuned Fasttext embeddings (+avg-fasttext) or the 768d pre-trained BioBERT [Lee et al., 2019] language model (+pool-biobert). For the document encoding, we use two LSTM layers (one forward, one backward) with 512 dimensions each, a discourse vector dense layer with 256 dimensions, L2 batch normalization and tanh activation. The discourse decoder is a 128-dimensional output layer with tanh activation and Huber loss. The network is trained with stochastic gradient descent over 50 epochs using the ADAM optimizer [Kingma and Ba, 2015] with a batch size of 16 documents, a learning rate of  $10^{-3}$ , 0.975 exponential decay per epoch, 0.0 dropout and  $10^{-4}$  weight decay regularization [Loshchilov and Hutter, 2017]. We chose these parameters using hyperparameter search on the WikiSection validation set. During training, we restrict the maximum document length to 396 sentences and maximum sentence length to 96 tokens, due to memory constraints on the GPU.

<sup>17</sup><https://lucene.apache.org>

Model all trained on Wikipedia	WikiSectionQA			MedQuAD			HealthQA		
	<i>R@1</i>	<i>R@10</i>	MAP	<i>R@1</i>	<i>R@10</i>	MAP	<i>R@1</i>	<i>R@10</i>	MAP
<i>Term-based models</i>									
TF-IDF	17.10	64.99	31.77	23.83	82.84	42.66	17.46	71.54	34.47
BM25	23.87	71.26	38.89	29.48	86.11	48.89	22.55	73.27	38.45
<i>Representation-based models</i>									
ARC-I	1.61	13.69	6.90	1.98	19.22	8.47	1.38	13.87	6.87
DSSM	22.82	74.31	39.02	13.11	55.92	27.38	10.50	46.44	22.04
C-DSSM	9.59	53.12	22.82	9.67	47.54	22.12	10.56	58.30	25.37
<i>Interaction-based models</i>									
ARC-II	10.38	53.62	23.61	9.19	47.58	21.66	11.26	58.85	26.09
DRMM	24.96	67.56	39.24	34.52	82.35	51.51	21.80	80.24	40.03
MatchPyramid	18.53	64.21	33.12	25.14	72.33	41.37	19.24	73.79	37.22
aNMM	4.77	32.17	14.03	7.15	37.18	17.08	3.74	27.20	12.07
KNRM	16.96	61.03	31.04	16.86	61.35	31.35	22.94	67.92	37.65
CONV-KNRM	34.36	77.25	48.72	42.70	84.54	57.57	33.13	85.41	50.55
HAR	45.31	84.15	58.38	<b>55.65</b>	<b>93.17</b>	<b>69.10</b>	43.20	88.34	58.80
<i>Combined models</i>									
Duet	18.34	59.13	31.74	20.50	65.91	35.28	17.27	64.81	32.13
MVLSTM	30.74	76.10	45.58	36.86	86.29	53.18	26.78	84.42	45.37
CDV+avg-glove	59.60	95.67	72.72	34.00	80.87	50.45	37.17	84.47	53.30
CDV+avg-fasttext	60.34	97.49	74.01	45.26	92.29	62.56	40.08	<b>89.80</b>	58.35
CDV+pool-biobert	<b>65.21</b>	<b>97.84</b>	<b>77.60</b>	39.96	91.32	58.91	<b>43.60</b>	88.12	<b>59.40</b>

TABLE 5.3: Experimental results for the Answer Passage Retrieval task on three Healthcare datasets. All models were trained using the self-supervised Wikipedia training set and applied without fine-tuning. Queries were evaluated by ranking 64 given candidates from the respective test sets. As queries we used  $\langle \text{entity}, \text{aspect} \rangle$  tuples in a representation suitable for the individual model.

The entity and aspect embeddings are trained with 128d Fasttext embeddings, followed by 128d BLSTM and dense embedding layers with tanh activations. The 1024d output layer is configured with sigmoid activation and BPMLL loss [Zhang and Zhou, 2006] to predict the Bloom hash ( $k = 5$ ) of the entity or aspect. The network is trained similarly to the CDV model, but we use 5 epochs with a batch size of 128 sentences, a learning rate of  $10^{-3}$  and 0.5 dropout.

### 5.4.2 Experimental Results

Table 5.3 shows the results on the Answer Passage Retrieval task using CDV and document matching models on three healthcare datasets. We observe that CDV consistently achieves significantly better results than all term-based, representation-based and combined models across all datasets. In comparison with pairwise interaction-based models, our representation-based retrieval model outperforms all tested models on average, scores best on WikiSection and HealthQA and second best on the MedQuAD dataset. Retrieval time per query is 247ms ( $\pm 43$ ms) on average. Figure 5.5 further shows that we correctly match between 67.5% and 91.4% of entities in the datasets and resolve 49.4% to 66.3% of all aspects.

**Comparison of model architectures.** Our query model is able to match most of the questions in the entity/aspect scheme (see Section 5.5 for exceptions). The results show that term-based TF-IDF and BM25 models can solve the healthcare retrieval task sufficiently with  $R@10 > 70\%$ . In contrast, none of the representation-based re-ranking models can achieve similar performance, except DSSM on WikiSectionQA. Most of the recent interaction-based and combined models outperform BM25 and have significant advantages on the MedQuAD dataset, which contains a large amount of generated information that can be matched exactly. We follow that simple word-level interactions are important for this task and representation-based models trade off this property for fast retrieval times.

**Background knowledge.** Our CDV model performs well on all data sets, but shows a significant advantage on the Wikipedia-based WikiSectionQA dataset. Although all models are trained on the same data, the only model with similar behavior is DSSM. One possible reason is that entity embeddings are an important source for background information, and these are mainly based on Wikipedia descriptions. 99.9% of the WikiSectionQA entities are covered in our embedding, 97.1% on MedQuAD and only 69.29% on HealthQA, because it does not only contain diseases. Sentence embeddings provide different levels of background knowledge and language understanding. The pre-trained GloVe embedding can handle the task well, but is outperformed by our fine-tuned Fasttext embedding and the large BioBERT language model.

**Domain adaptability and task coverage.** Figure 5.6 shows the performance of our CDV+avg-fasttext model across all data sources, most of them contained in MedQuAD. This distribution reveals that our model top-1 accuracy is stable in the adaptation to most sources except National Cancer Institute (CancerGov). However, we notice that  $R@10$  performance is high among all sources except SeniorHealth. Figure 5.7 shows that  $R@10$  performance across the most frequent aspects is over 93% in most cases, but with varying top-1 recall. We will address these errors in Section 5.5.

**Impact of contextual dependencies.** Score predictions in CDV are calculated on sentence level with respect to long-range context across the entire document. In Figure 5.4, we observe that the model is able to predict the entity (top curve) consistently over the document, although there are many coreferences in Wikipedia text. The aspect curve (center) clearly shows the beginning of the expected section Symptoms and the model is uncertain for the following sentences. Finally, the average score (bottom curve) shows a coherent prediction.

## 5.5 Discussion and Error Analysis

We perform an error analysis on the predictions of the CDV+avg-fasttext model to identify main reasons for answer misranking. For this purpose we analyse samples in which the model ranks a wrong passage at the top-1 position. We look at 50 random mismatched samples per dataset to understand the individual challenges per source. We now discuss the main findings.

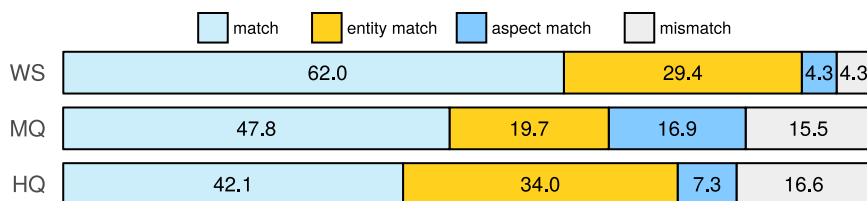


FIGURE 5.5: Entity/aspect matching (values in percent) observed on all examples in the three evaluation datasets.

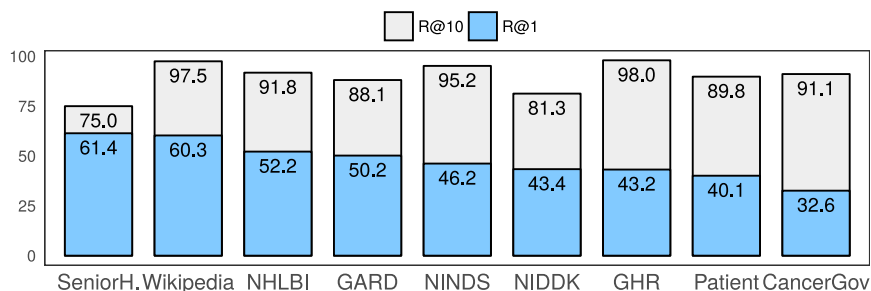


FIGURE 5.6: R@1 and R@10 performance of the CDV-EA+pool-biobert model across all data sources. All sources except Wikipedia and Patient are contained in MedQuAD dataset.

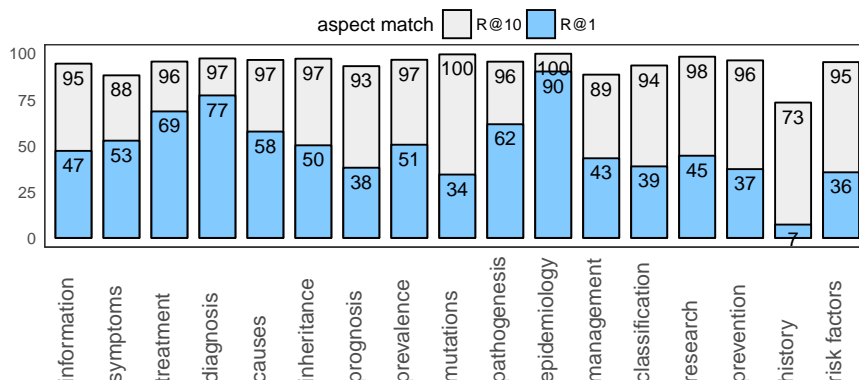


FIGURE 5.7: R@1 and R@10 prediction performance of the 17 most frequent aspects in all test sets (of 34 total). Aspects are sorted from left to right by frequency.

**Related entities.** Figure 5.8 shows that a main source of entity errors comes from selecting passages that belong to related entities. This includes entities that are superclasses or subclasses of the gold truth, e.g. selecting a passage covering “Diarrhea” when “Chronic Diarrhea in Children” is the query entity. These errors are most significant in WikiSectionQA and MedQuAD, because HealthQA covers mostly common diseases. We especially observe this in samples from Genetic Home Reference and National Cancer Institute. Figure 5.6 shows that R@1 is low for samples from these sources, whereas their R@10 is high. That is because genetic conditions and cancer types inherently contain entities with very similar names and descriptions. For instance, we see “Spastic Paraplegia Type 8” falsely resolved to “Spastic Paraplegia Type 11”. As the representations are close to each other in vector space, the correct samples are almost always found within the top-10 ranked candidates, corresponding with the high R@10.

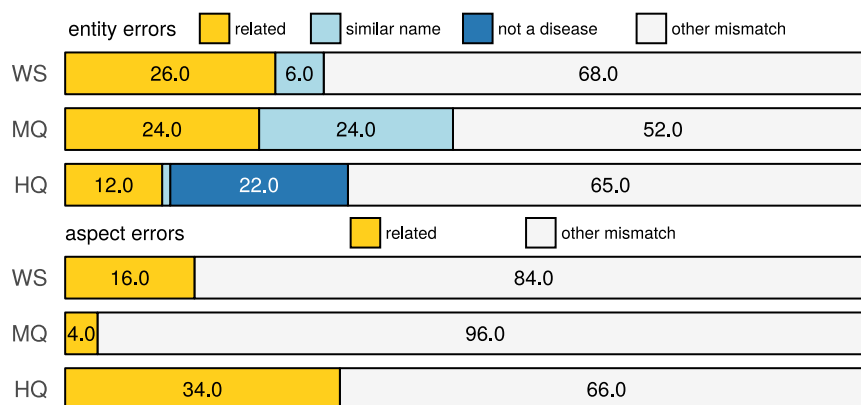


FIGURE 5.8: Error classes for entity and aspect mismatch (values in %) from manual analysis of 150 mismatched queries.

**Related aspects.** Likewise, we observe that in HealthQA 34% and in WikiSectionQA 16% of aspects are mismatched to related aspects. Figure 5.7 shows the distribution of aspects and the model’s ability to resolve them. It is salient that some aspects are especially difficult to resolve. Aside from the fact that these aspects are in the long tail, a further analysis reveals that they are often resolved to related aspects. For example, passages covering classification are often very similar and therefore confused with passages about diagnosis and symptoms. The same holds true for prognosis and management. Queries asking for prevalence of a disease are often resolved to information passages, because disease frequency is often mentioned in these introductory texts. In general, passages about related aspects often share similar tokens and document context, which makes their distinction more difficult.

**Out-of-scope questions.** 25% of queries from HealthQA contain entities that are no diseases but procedures, drugs or other entity types. As our model is trained on textual data covering diseases only, we do not expect it to fully resolve these entities. However, we observe that the model is capable of finding the correct passage for 23% of unseen entities. This shows that while our model is not trained on such entity types, the fallback embedding described in 5.2.1 still allows to generalize even to non-diseases in these cases.

**Evaluation vs. real-world application.** We further identified a number of errors related to the structure of the evaluation, that would be less problematic or even beneficial in real-world application. The model frequently ranks passages to the top which answer the query but have a differing aspect assigned. We observe this in 24% of analysed samples from WikiSectionQA and 18% from HealthQA. This often seems to be caused by the non-discrete nature of topical aspects. In practice a passage can cover more than one aspect, but our evaluation does currently not capture this ambiguity. Additionally we find some mismatches between passages and their ground-truth aspects, which can be ascribed to writing errors in WikiSectionQA and labeling errors in HealthQA. Aspects in MedQuAD are less ambiguous in general and only

4% fall into this error class. Figure 5.5 shows that the model therefore resolves more aspects correctly for MedQuAD queries.

**Irrelevant text within passage boundaries.** Another finding is that 28% of analysed samples from the MedQuAD dataset contain boilerplate text unrelated to a specific entity. The boilerplate includes repeated text such as information about how data was collected. In this case our model is able to detect relevant parts of a passage (see Figure 5.4), but the remaining irrelevant sentences lead to a worse ranking of the passage. Evaluating with flexible passage boundaries would eliminate this issue and be a better match for real-world scenarios, in which the interest of a medical professional is mainly focused on non-boilerplate parts of a document.

**Complex questions.** We find that most questions in our evaluation can be represented as tuples of entity and aspect without information loss. However, in 4% of analysed queries in the HealthQA dataset we see a mismatch between question and query. For instance, the question “How common is OCD in Children and Young People?” which is more specific than the assigned query tuple “Obsessive-compulsive disorder” and prevalence. Different solutions are possible for representing more complex queries, e.g. by composing multiple queries during retrieval. We leave these questions for future research.

## 5.6 Related Work

There is a large amount of work on *Question Answering* (QA) [Seo et al., 2017; Wang et al., 2017b], also applied to healthcare [Abacha et al., 2019; Jin et al., 2019] which focuses primarily on factoid questions with short answers. Typically, these models are trained with labeled question-answer pairs. However, it was shown that these models are not suitable for extracting local aspects from long documents, and especially not for open-ended, long answer passages [Tellex et al., 2003; Keikha et al., 2014; Nanni et al., 2018; Zhu et al., 2019]. We therefore frame our task as a *passage retrieval* problem, where the system’s goal is to extract a concise snippet (typically 5–20 sentences) out of a large number of long documents. Furthermore, following studies from EBM [Richardson et al., 1995; Cheng, 2004; Huang et al., 2006], we focus on *structured healthcare queries* instead of free-text questions.

**Discourse-aware representations.** Recently, new approaches have emerged that represent local information in the context of long documents. For example, Cohan et al. [2018] approach the problem as abstractive summarization task. The authors use hierarchical encoders to model the discourse structure of a document and generate summaries using an attentive discourse-aware decoder. In our prior work on SECTOR [Arnold et al., 2019], we apply a segmentation and classification method to long documents to identify coherent passages and classify them into 27 clinical aspects. The model produces a continuous topic embedding on sentence level using BLSTMs, which has similar properties to the micro-topics described earlier by Arora et al. [2016] as *discourse vector* (“what is being talked about”).

We follow these ideas as the groundwork for our approach. Our proposed model is based on a hierarchical architecture to encode a continuous discourse representation. To the best of our knowledge, our model is the first to use discourse-aware representations for answer retrieval. Additionally, we address the problem of sparse training data and propose a multi-task approach for training the model with self-supervised data instead of labeled examples.

**Passage matching.** A baseline approach to the passage retrieval problem is to split longer documents into individual passages and rank them independently according to their relevance for the query. Passage matching has been done using term-based methods [Robertson and Jones, 1976; Salton and Buckley, 1988], most prominently in TF-IDF [Jones, 1972] or Okapi BM25 [Robertson et al., 1995]. However, these methods usually do not perform well on long passages or when there is minimal word overlap between passage and query. Therefore, most neural models tackle vocabulary mismatch using semantic vector-space representations.

*Representation-based matching models* aim to match the continuous representations of queries and passages using a similarity function, e.g. cosine distance. This can be done on sentence level (ARC-I [Hu et al., 2014]), which does not work well if queries are short and passages are longer than a few sentences. Therefore, most approaches learn distinct query and passage representations using feed-forward (DSSM [Huang et al., 2013]) or CNN convolutional neural networks (C-DSSM [Shen et al., 2014]).

*Interaction-based matching models* focus on the complex interaction between query and passage. These models use CNNs on sentence level (ARC-II [Hu et al., 2014]), match query terms and words using word count histograms (DRMM [Guo et al., 2016]), word-level dot product similarity (MatchPyramid [Pang et al., 2016]), attention-based neural networks (aNMM [Yang et al., 2016a]), kernel pooling (K-NRM [Xiong et al., 2017]) or convolutional n-gram kernel pooling (Conv-KNRM [Dai et al., 2018]). Eventually, Zhu et al. [2019] utilize hierarchical attention on word and sentence level (HAR) to capture interaction of the query with local context in long passages.

While interaction-based models can capture complex correlations between query and passage, these models do not include contextualized local information—e.g. long-range document context that comes before or after a passage—which might contain important information for the query. To overcome this problem, Mitra et al. [2017] combine document-level representations with interaction features in a deep CNN model (Duet). Wan et al. [2016] utilize BLSTMs (MVLSTM) to generate positional sentence representations across the entire document.

We combine the representation approach with interaction. Our proposed model is able to learn the interaction between the words of the passage and the discourse using a language model. At the same time, it encodes fixed sentence representations that we use to match query representations. Consequently, our model does not require pairwise inference between all query–sentence pairs, which is usually circumvented by re-ranking candidates [Gillick et al., 2018]. Instead, our model requires only a single pass through all documents at index time.

Furthermore, by encoding discourse-aware representations, the model is able to access long-range document context which is normally hidden after the passage split. We compare our approach to all the discussed matching models in Section 5.4.

## 5.7 Conclusions

In this chapter, we have approached RQ 3, the embedding of discourse structure into document representations. We presented CDV, a contextualized document representation that is encoded by applying Neural Machine Reading to long documents without external supervision. The model builds upon our previous results on RQ 1 and 2 by extending sentence representations with complementary distributed information from entity and aspect embeddings. We have proposed to integrate pre-trained embeddings using a multi-task objective function. This approach retains the properties of the embedding spaces and helps the model to generalize over unseen examples. We further approached RQ 4, the retrieval of answer passages from domain-specific text resources. We showed the applicability of the CDV model to three healthcare answer retrieval tasks with best or second best accuracy compared to 14 strong baseline models. We showed that in comparison to previous approaches, CDV is able to integrate structural document context into its representations, which helps to resolve long-range dependencies normally not visible to passage re-ranking models. Our CDV representations can be precomputed and used to retrieve passages for ad-hoc queries with efficient k-nearest neighbor search. Furthermore, we showed that by integrating a biomedical language model as lowest layer, accuracy improves significantly. Eventually, our CDV model can be trained with self-supervised data from medical Wikipedia articles and applied to different domains, such as biomedicine or consumer healthcare. In summary, we have shown that CDV fulfills all properties that we introduced for our vision of automatic language understanding in Section 1.2. This makes it an appropriate model to satisfy our hypothesis of Neural Machine Reading. In the next chapter, we discuss systems that utilize CDV and other building blocks we introduced in this thesis for applications of human information seeking.



## Chapter 6

# Systems

In this chapter, we present four implementations of our Neural Machine Reading architectures in form of system applications: TASTY is a text editor that utilizes our Entity Linking components to interactively support human information-seeking intentions (Section 6.1). TraiNER is a system to efficiently train Named Entity Recognition models with active learning (Section 6.2). SMART-MD is a clinical decision support system that utilizes TASTY and SECTOR to find passages in research articles (Section 6.3). CDV Healthcare Retrieval utilizes CDV for finding answers for clinical questions in large corpora (Section 6.4). All these systems are implemented in Java using the *TeXoo Text Extraction Framework* which we release as open-source<sup>1</sup>.

### 6.1 TASTY: Interactive Editor for Entity Linking As-You-Type

We introduce TASTY (Tag-as-you-type)<sup>2</sup>, a novel text editor for interactive Entity Linking as part of the writing process<sup>3</sup>. TASTY supports the author of a text with complementary information about the mentioned entities shown in a ‘live’ exploration view. The system is automatically triggered by keystrokes, recognizes mention boundaries and disambiguates the mentioned entities to Wikipedia articles. The author can use seven operators to interact with the editor and refine the results according to his specific intention while writing. Our implementation captures syntactic and semantic context using a robust end-to-end LSTM sequence learner and word embeddings. We demonstrate the applicability of our system in English and German language for encyclopedic or medical text. TASTY is currently being tested in interactive applications for text production, such as scientific research, news editorial, medical anamnesis, help desks and product reviews.

#### 6.1.1 Design Challenges

Entity Linking is the task of identifying mentions of named entities in free text and resolving them to their corresponding entries in a structured knowledge base [Hachey et al., 2013].

<sup>1</sup>TeXoo is available at <https://github.com/sebastianarnold/TeXoo> under Apache V2 license.

<sup>2</sup>This system was published by S. Arnold, R. Dziuba, and A. Löser [2016a]. “TASTY: Interactive Entity Linking As-You-Type”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 111–115.

<sup>3</sup>A live demo is available at <https://tasty.demo.dataxis.com>

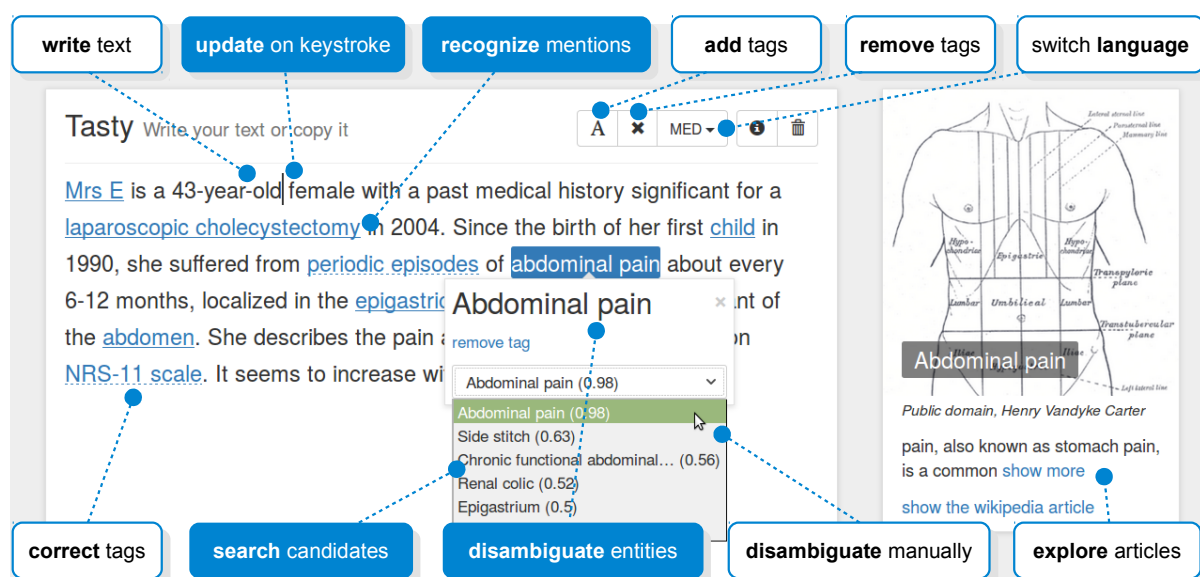


FIGURE 6.1: Example of writing a text in TASTY's user interface. Named entities are displayed as tags, articles appear on the right side. White boxes denote interaction operators, filled boxes show system actions.

These two steps are often executed as batch process *after* the document has been written by the author. Contrary, doctors during a medical anamnesis, technicians writing supportive manuals or assistants in help desks desire Entity Linking *during* writing. Ideally, a machine could highlight relevant information about recognized entities while the author is typing the text and gradually adapt the results to complement his task.

TASTY is such a novel text editing interface for fine-grained tagging of articles as part of the writing process. Figure 6.1 shows an example of the editor in use. While the author is typing, a contextual sequence learner immediately recognizes mention boundaries, tags them in-line, resolves associated articles and displays them beside the document. When more context is written, the system reacts and refines boundaries and associations without interrupting the process. The author can *add*, *remove* and *disambiguate* tags according to his task and knowledge. TASTY's extraction model recognizes multi-word mentions and identifies entities that are both in and outside the knowledge base. It does not require linguistic features, can be applied to multiple languages without hyperparameter changes and is robust to misspelled or out-of-vocabulary words. To our knowledge, TASTY is the first system that implements interactive Entity Linking for manifold scenarios.

**TASTY supplies doctors with supplemental materials.** As demonstration example we showcase a medical *History and Physical Examination (H&P)* write-up, where doctors write text about a patient's history and conditions. TASTY can recognize these medical entities and link them to Wikipedia articles. Other possible targets are e.g. research papers or doctor letters. As a result, a doctor may learn from these documents additional insights for sharpening her focus.

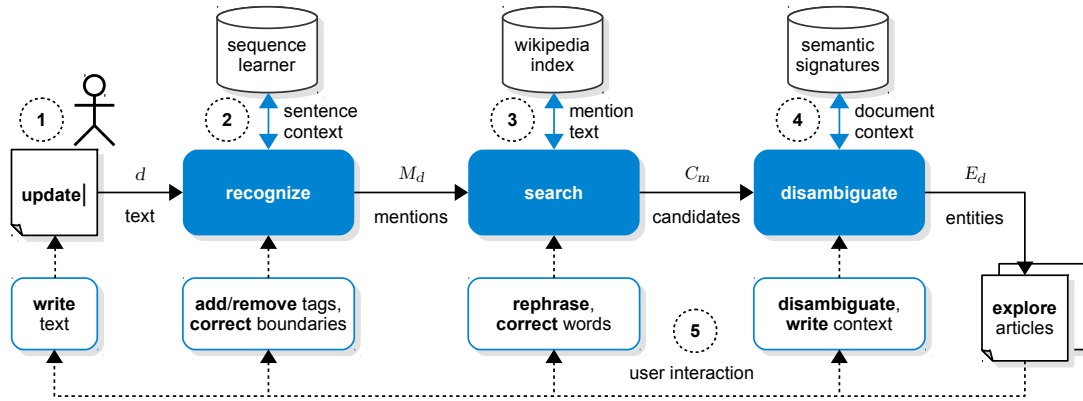


FIGURE 6.2: Overview of the interactive Entity Linking process in TASTY. While the author is writing a text, the system recognizes mentions, searches for entity candidates and disambiguates the mention to its corresponding Wikipedia article. The author is able to interact with every stage of the extraction process.

We showcase the following scenario as an example H&P (see Figure 6.1): The doctor starts by writing the first sentence about her patient: “Mrs E is a 43-year-old female with a past medical history significant for a laparoscopic cholecystectomy”. TASTY responds to key strokes, recognizes mentions, searches for candidates and displays a complementary article for cholecystectomy next to the document. The doctor might explore the article and incrementally learn about important aspects of this condition. She might continue writing “she suffered from periodic episodes of abdominal pain localized in the epigastric region” and manually select a more precise disambiguation for the phrase “abdominal pain”. She may correct further tagging errors, e.g. remove the unwanted tag Mrs E. In case of a missing tag, the doctor can edit a phrase, e.g. “NRS-11 pain scale” and tag it manually. The system reacts and returns a corresponding disambiguation.

### 6.1.2 Interactive Entity Linking Process

We implement interactive Entity Linking using *mention recognizer*, *candidate searcher* and *disambiguator* stages [Hachey et al., 2013]. We extend the process by an interactive cycle that includes *partial update* and *user feedback* operators, as shown in Figure 6.2. We demonstrate TASTY in English (EN) and German language (DE) and for a specialized medical scenario (MED).

**Step 1: Update while the author is typing.** TASTY’s user interface is based on a lightweight rich text editor<sup>4</sup> that we extend to display named entity mentions as in-line tags. TASTY captures the author’s key strokes and detects word boundaries after space or punctuation characters. We split a document of length  $n$  into a sequence of word tokens  $d = (w_1, \dots, w_n)$  using a language-independent whitespace tokenizer<sup>5</sup>. In a partial update step, we analyze only the changed portion  $\tilde{d} = (w_b, \dots, w_e)$ ,  $1 \leq b < e \leq n$  of the document. We expand indexes  $b$  and  $e$  to sentence boundaries and omit any further linguistic processing.

<sup>4</sup>We use Quill v1.0.0-beta.11 <http://quilljs.com>

<sup>5</sup>We use PTBTokenizer from Stanford CoreNLP 3.6.0 <http://stanfordnlp.github.io/CoreNLP/>

**Step 2: Recognize mention boundaries.** We define a mention  $m$  as the longest possible span of adjacent tokens that refers to an entity or relevant concept of a real-world object, such as “epigastric region”. In TASTY, we further assume that mentions are non-recursive and non-overlapping. The objective of this step is to detect all mention spans  $M_{\tilde{d}} = \{m_i\}$  in the document portion. We model this task as context-sensitive sequential word labeling problem. We predict for each token  $w_t \in \tilde{d}$  a target label  $\hat{y}_t$  according to the BIOES tagging scheme [Ratinov and Roth, 2009] with respect to its surrounding words (Eq. 6.1). From these labels, we populate  $M_{\tilde{d}}$  in a single iteration. For the prediction task, we utilize Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997], which are able to capture long-range sequential context information with short answer times. The input is a sequence of word feature vectors  $\mathbf{x}(w_t)$  with three components: First, we use lowercase letter-trigram word hashing [Huang et al., 2013] to encode word syntax on character level. This technique splits a word into discriminative three-letter ‘syllables’ with boundary markers, e.g. `cell`  $\rightarrow$  `{#ce, cel, ell, ll#}` to make the bag robust against misspellings and out-of-vocabulary words. Second, we utilize word embeddings [Mikolov et al., 2013a]<sup>6</sup> to represent word semantics in dense vector space. Third, we encode surface form features by generating a vector of flags that indicate e.g. initial capitalization, uppercase, lower case or mixed case.

$$\hat{y}_t = \arg \max_{l \in \{B, I, O, E, S\}} p(\mathbf{y}_t = l \mid \mathbf{x}(w_b), \dots, \mathbf{x}(w_{t-1}), \mathbf{x}(w_t), \mathbf{x}(w_{t+1}), \dots, \mathbf{x}(w_e)) \quad (6.1)$$

We pass through  $\tilde{d}$  bidirectionally using a stacked BLSTM+LSTM architecture [Arnold et al., 2016b]<sup>7</sup>. Our recognition component can be trained ‘end-to-end’ with only few thousand labeled sentences. For the demonstration, we provide three different pre-trained models: EN is trained to recognize named entities (persons, organizations, locations and misc) in English encyclopedic text, DE captures proper nouns (untyped) in German encyclopedic text, and MED recognizes biomedical terms in scientific text.

**Step 3: Search for candidate links.** Our next step is to resolve a subset of Wikipedia article candidates  $C_m$  for each of the detected mentions  $m$ . We especially aim to capture a large number of candidates for highly ambiguous terms such as scale or child. For this task, we create an index of 4.5M English and 1.6M German Wikipedia abstracts<sup>8</sup>. We use redirects and anchor phrases to capture alternative writings and synonyms [Hachey et al., 2013]. We apply a dictionary-based technique described by Ling et al. [2015] and query the index for candidates  $C_m = \{c_j \mid \forall m \in \tilde{d} : c.\text{title} \approx m.\text{span}\}$  using phrase queries with BM25 similarity<sup>9</sup> for retrieval. In case of an empty result, we return NIL (non-linkable entity).

<sup>6</sup>We use a 150-dimensional lowercase word2vec model trained on English and German Wikipedia.

<sup>7</sup>We implement the network using Deeplearning4j 0.6.0 <https://deeplearning4j.org>

<sup>8</sup>We use DBpedia version 2015-10 <http://wiki.dbpedia.org/datasets>

<sup>9</sup>We use the implementation in Lucene 6.1.0 <http://lucene.apache.org>

**Step 4: Disambiguate associated articles.** From the set of candidates  $C_m$ , we want to pick the most likely entity associations  $E_d = \{(m_i, \hat{c}_j)\}$ . We do this by picking the candidate  $\hat{c}$  with maximum score depending on the mention and current document context:

$$\hat{c} = \arg \max_{c \in C_m} \text{score}(c \mid m, d) \quad (6.2)$$

As scoring function, we utilize short text similarity [Kenter and de Rijke, 2015] between mention context  $m.d$  and a candidate article  $c.d$ . We utilize word embeddings to calculate vectors  $\mathbf{x}_{\text{emb}}(w_t)$  for every token in the document and aggregate them into a normalized mean document vector that we use as semantic signature  $\mathbf{s}(d)$ :

$$\mathbf{s}(d) = \frac{1}{n} \sum_{w_t \in d} \mathbf{x}_{\text{emb}}(w_t) \quad (6.3)$$

We finally use cosine similarity between the semantic signatures as scoring function:

$$\text{score}(c \mid m, d) = \frac{\mathbf{s}(m.d) \cdot \mathbf{s}(c.d)}{\|\mathbf{s}(m.d)\| \|\mathbf{s}(c.d)\|} \quad (6.4)$$

**Step 5: Feed back user interaction.** TASTY offers seven feedback operators that enable an author to interact with every component in the extraction process. All operators are based on typing or text selection. Using *write*, the author emits more context and the system reacts to word boundaries by triggering a partial update. He might also *rephrase* single words, triggering the system to update surrounding annotations. Using the *add* button, he is able correct false negative predictions from the recognition component. The system will tag the selected mention, generate candidates and decide for an associated article. The *remove* button deletes selected tags to correct false positive predictions. The author can *correct* boundaries of an existing tag, and the system will update the link if necessary. If the boundaries of a tag are correct, but the link is not, he can *disambiguate* by assigning a different candidate from the drop-down menu. Finally, the author benefits from several operators to *explore* the articles. Corrections are directly executed in the local session and fed back as training data to our model.

### 6.1.3 Application Scenarios

We showcased TASTY’s editor with pre-trained models to 21 experienced professionals and learned about exciting application scenarios which are shown in Table 6.1. A large group of users applied the results of in-line Entity Linking to subtasks with *exploratory search intention* [Marchionini, 2006]: *look up* facts or definitions for entities in the text, *learn* from complementary articles, *compare* written text against text in archives, *verify* information, *integrate* with existing tagging schemes. For future implementations, users suggested the application of *investigatory* subtasks: *evaluate* text to fit a desired tone or vocabulary, *discover* alternatives or get *advice* from user reviews or experts. We aim to extend TASTY with specialized models for these scenarios in future work.

Scenario	Example	Subtasks	
<b>Research</b>	report writing	pin topics find sources	lookup explain
<b>Editorial</b>	news authoring	annotate paragraphs identify topics and tags	style suggestion search engine optimization
<b>Diagnosis</b>	anamnesis	lexicon search patient history	side effects medical compatibility
<b>Help Desk</b>	customer support	FAQ search related tickets	manuals expertise search
<b>Shopping</b>	product order	price comparison feature infobox	user reviews purchase advice

TABLE 6.1: Five example scenarios for TASTY’s application.

## 6.2 TraiNER: Bootstrapping Named Entity Recognition

Named Entity Recognition (NER) is an important preprocessing step for downstream tasks such as Named Entity Linking, Relation Extraction or Question Answering. These tasks depend on high NER recall [Pink et al., 2014]. However, often pre-trained NER models are used in scenarios where they do not exactly fit the underlying task. For example, a system that links text with a product database, such as cars or car parts, can utilize a fine-grained NER model for products or technologies [Ling and Weld, 2012]. A more accurate solution requires a NER model specifically trained for cars, but this step usually requires a large set of annotated text as training data. In corporate scenarios, this data does not exist, but instead lists of examples, e.g. car brands and model names, are stored in databases.

We propose TraiNER<sup>10</sup>, an adaptive entity extractor that can be trained with only a dictionary of seed examples and a collection of unlabeled documents as training data. We bootstrap a model using the seeds, sample training examples from the data and ask an expert user to refine the labeling task by manual annotation. We use this feedback to generate additional training examples and fit a personalized model after few iterations.

### 6.2.1 Active Learning Process

In our scenario, we train the extractor with the following assets: A corpus of *unlabeled text* from the target domain. A *seed list* of keywords that express possible entity instances. An *expert* who knows the task and is able to complete a set of labeling tasks. Optionally, a small labeled set with *evaluation data*. We utilize these assets to build a model using an active learning process [Settles, 2010], which is visualized in Figure 6.3.

**1. Bootstrapping training data from dictionaries.** We start to bootstrap training data by collecting a seed list  $\mathcal{S}$  of entity names from a database, e.g. Wikidata. We utilize an efficient implementation of Backward-Oracle-Matching algorithm [Faro and Lecroq, 2009] to match all

<sup>10</sup>This system is based on unpublished work by S. Arnold, R. Schneider, C. Kümmel, R. Mehlitz, T. Oberhauser and A. Löser.

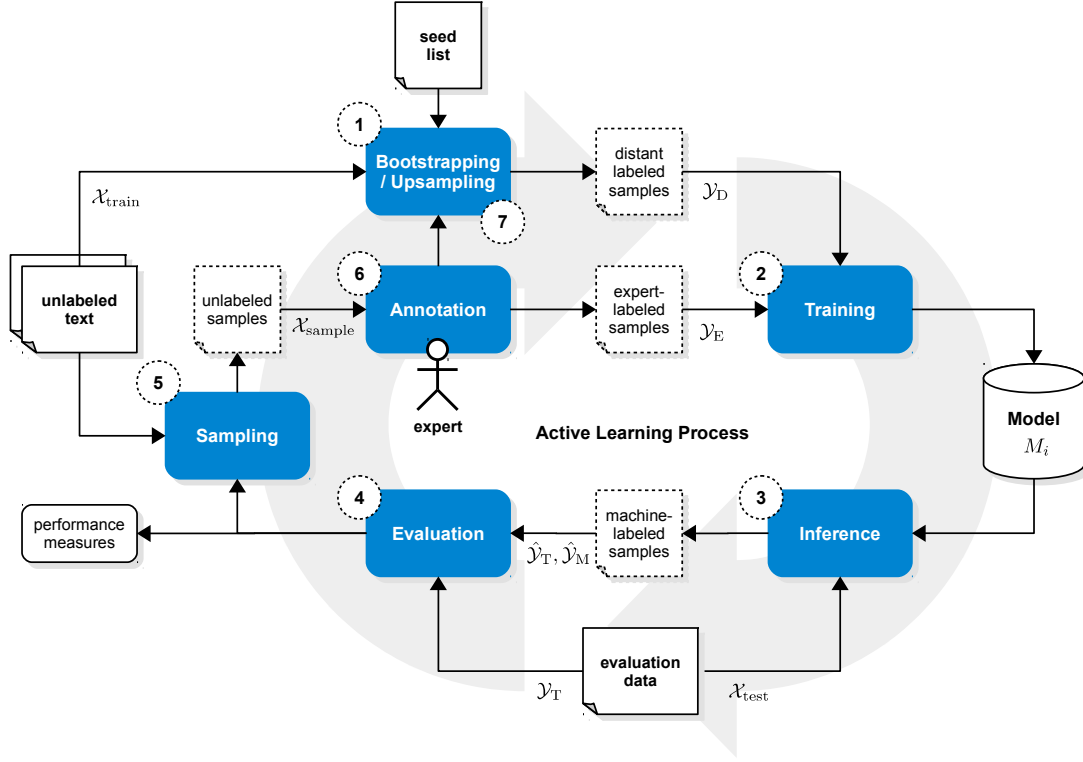


FIGURE 6.3: TrainNER active learning process.

entries of  $\mathcal{S}$  to the documents of the unlabeled text corpus. We use lowercase matching of all terms of length 3 or longer. We discard matches which do not align with token boundaries entirely and give priority to longest matches. The resulting data set contains sentences with sparse distant-supervised labels  $\mathcal{Y}_D = \{(s, \text{match}(s, \mathcal{S})) \mid s \in \mathcal{X}_{\text{train}}\}$ . Because the dictionary is incomplete, the data set may contain a large number of false negatives. The set may also contain false positive mentions and wrong boundaries, resulting from morphological similarity.

**2. Training an efficient NER model.** We utilize our TASTY sequence labeler with robust encodings and contextual embeddings [Arnold et al., 2016b] that can be trained end-to-end. We call this model  $M_i$  at iteration  $i$ . For bootstrapping, we optimize  $\tilde{y} = M_0(s) \forall (s, \tilde{y}) \in \mathcal{Y}_D$  for a random sample from the training data. We empirically observed that this step requires at least 4,000 training sentences with high variance to produce a well-performing model. As the process continues, we call this step iteratively to retrain  $M_{i+1}$  with improved labels.

**3. Inference on evaluation data.** We apply  $M_i$  to a labeled test dataset  $\mathcal{Y}_T = \{(s, y) \mid s \in \mathcal{X}_{\text{test}}\}$  to produce labels for evaluation:  $\hat{\mathcal{Y}}_T = \{(s, M_i(s)) \mid s \in \mathcal{X}_{\text{test}}\}$ . Furthermore, we produce labels for a random sample of machine-labeled examples  $\hat{\mathcal{Y}}_M = \{(s, M_i(s)) \mid s \in \mathcal{X}_{\text{train}}\}$ .

**4. Evaluation of current iteration.** We evaluate predictions  $\hat{\mathcal{Y}}_T$  with respect to the ground truth  $\mathcal{Y}_T$  using  $F_1$  measure. We stop after this step if the results are above a given threshold or the annotation budget is reached.

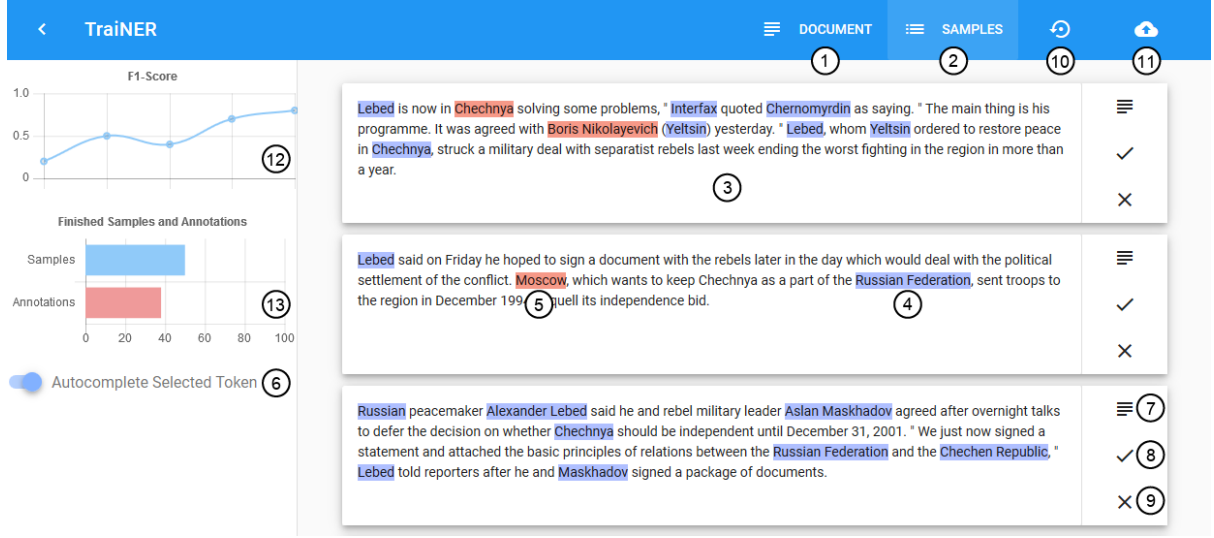


FIGURE 6.4: Screenshot of the TraiNER user interface for human annotation.

**5. Sampling examples for manual annotation.** From machine-labeled predictions  $\hat{\mathcal{Y}}_M$ , we pick 100 examples  $\mathcal{X}_{\text{sample}}$  that require manual examination. We calculate a confidence score for each prediction using the maximum class probability from the NER softmax. We use this value for uncertainty sampling [Settles, 2010].

**6. Annotation by human expert.** The human oracle is asked to annotate the 100 samples  $\mathcal{X}_{\text{sample}}$  with expert labels  $\mathcal{Y}_E = \{(s, \text{label}(s)) \mid s \in \mathcal{X}_{\text{sample}}\}$ . We use a graphical annotation interface, which we discuss in the next section.

**7. Upsampling human annotations to generate training data.** The number of examples required to improve the NER model is often too large for a human task. We therefore utilize an upsampling method to generate more training samples from the 100 expert labels. We apply the dictionary bootstrapping method to a random sample of the training data with all new entity names contained in the expert labels:  $\mathcal{Y}_{D+} = \{(s, \text{match}(s, \mathcal{S} \cup \mathcal{Y}_E)) \mid s \in \mathcal{X}_{\text{train}}\}$ .

**Iterative loop.** We continue with step 2 and optimize the next generation model  $M_{i+1}$  with expert and distant labels:  $\tilde{y} = M_i(s) \forall (s, \tilde{y}) \in \mathcal{Y}_E \cup \mathcal{Y}_{D+}$ .

## 6.2.2 Graphical User Interface

The TraiNER interface is optimized for annotating named entity mentions with very few required clicks. Figure 6.4 shows the interface with its operators. The system can be used in two scenarios: An expert can use it in *offline mode* (1) to exhaustively annotate a complete data set with named entity mentions. Or she can use it in *online mode* (2) and iteratively annotate small batches of sampled sentences, as described in the previous section. Each text snippet is displayed as an interactive text box (3). Bootstrapped model annotations are shown in blue (4),



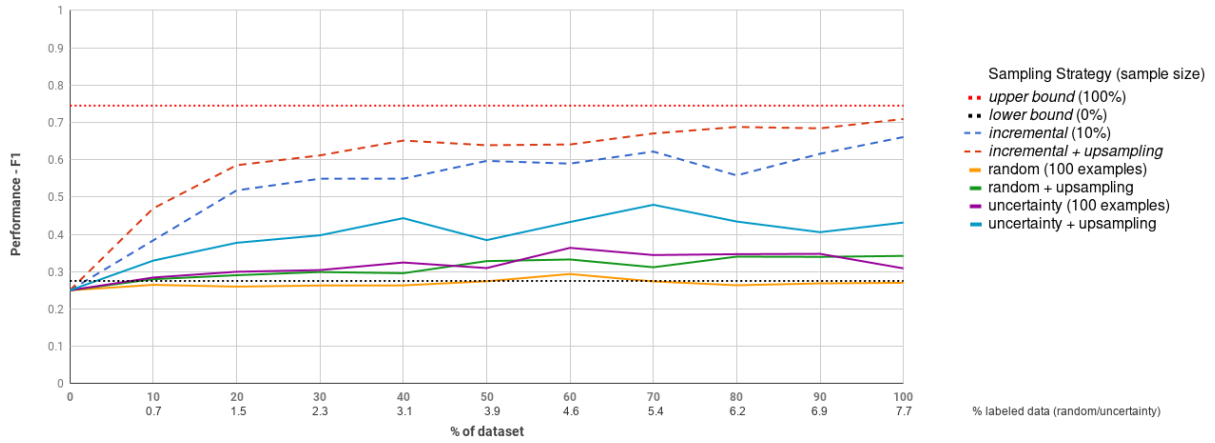


FIGURE 6.5: Comparison of TraiNER sampling strategies.  
(Figure adapted from Kümmel [2018])

user corrections are shown in red (5). By selecting the text with the mouse cursor, the user can efficiently add, delete and correct annotations. An autocompletion operator helps to quickly annotate entire words with single clicks (6). The user can also display the snippet in the context of the entire document (7). The user needs to accept the annotations for the snippet (8) or reject all of them (9). After this operation, the snippet is closed. The user can reopen the snippets later for review or correction (10). After annotation, the user sends the annotations back to the model to start a training iteration (11). The evaluation scores are displayed after each iteration (12). To gain an overview on the annotation progress, the number of remaining samples and annotations are displayed for the user (13). In our experiments with ten expert annotators using the system for 8 hours each, we observed that an average annotator achieved to label 305 sentences per hour.

### 6.2.3 Experimental Results

We simulated the online labeling scenario on the *i2b2* clinical concept recognition dataset [Uzuner et al., 2011]. We used a seed list with 24,125 entity names from Wikidata for bootstrapping and a 20% test split of the dataset as evaluation set. We conducted several training runs with 10 iterations each: *random sampling* and *uncertainty sampling* of 100 human-labeled examples, *incremental sampling* of all examples (10% of the training data per iteration), both with and without *upsampling*. The training curves are shown in Figure 6.5. From training the NER model on the fully labeled training set we observe an upper bound performance of 74.5%  $F_1$ . Using only the bootstrapping approach yields a lower bound performance of 27.5%  $F_1$ . We further observe that incremental sampling requires 50% of the training data to reach 60%  $F_1$ , while upsampling reduces these costs to 25% of the data. With random sampling, the model does not significantly improve after 10 iterations (7.7% of the training data), even with upsampling. Uncertainty sampling with upsampling can speed up this process, so that the model performs with 45%  $F_1$  or better after 5% of the training data has been labeled.

### 6.3 Smart-MD: Clinical Decision Support System

Medical doctors, in particular at emergencies, often need to make fast decisions and without studying the latest research results from journals thoroughly. In particular less experienced doctors might overlook alternative treatments or therapies and often fall back to potentially less effective standard procedures known from their academic studies. Despite the fact that most queries of doctors are of informational intent [Yoo and Mosa, 2015; White and Horvitz, 2014], standard medical search engines, like PubMed<sup>11</sup>, still focus on filtering documents using keyword queries. Ideally, a doctor could use an effective search engine for retrieving diverse and potentially unknown results from the latest literature about symptoms, therapies, medications, treatments or other often requested aspects during the anamnesis.

#### 6.3.1 Demonstration Scenario

Consider the case of a doctor searching for treatments of *Lyme disease*, an infectious disease caused by bacteria of the *Borrelia* type which is mainly spread by *ticks*. She will study essential articles and will find the transmission of ticks from birds to humans as main cause. While she knows from her academic studies that antibiotics such as *doxycycline* will help most patients, she might oversee that certain patients with cardiac diseases will likely suffer from this treatment and should rather be treated with *ceftriaxone*-based antibiotics. Ideally, the system would retrieve all treatments for Lyme disease and would display an aggregated overview of different treatments, including some paragraphs of text which explain infrequent edge cases.

We demonstrate SMART-MD<sup>12</sup>, an IR system that provides such a functionality for medical professionals<sup>13</sup>. The system takes as input diseases and a list of optional topical aspects. Figure 6.6 shows a typical result for the query “lyme treatment” (1). Given the query, the system retrieves two highly relevant paragraphs about treatments from two articles on Lyme disease or on *Borrelia* (4). It recognizes and aggregates important facets in these paragraphs, such as correlating medical terms or topics and provides the user these facets for query refinement (2). Furthermore, SMART-MD shows a distribution of treatments (3) and the user can narrow the query to a particular novel and previously unknown treatments. Finally, the user may click on an interesting paragraph to inspect the context of the entire document (5). Thereby the system highlights the topic of each relevant paragraph (6). In particular with long documents, this fine granularity at paragraph level permits the reader to skip many irrelevant passages.

#### 6.3.2 Passage Retrieval Process

SMART-MD is built upon two neural information extractors which process the dataset at load time. The *topic extractor* assigns a distribution of topics to each sentence in the dataset. The

<sup>11</sup><https://www.ncbi.nlm.nih.gov/pubmed/>

<sup>12</sup>This system was published by R. Schneider, S. Arnold, T. Oberhauser, T. Klatt, T. Steffek, and A. Löser [2018]. “Smart-MD: Neural Paragraph Retrieval of Medical Topics”. In: *The Web Conference 2018 Companion*. IW3C2, pp. 203–206.

<sup>13</sup>A video is available at <https://www.youtube.com/watch?v=kcDi7qQxpBo>

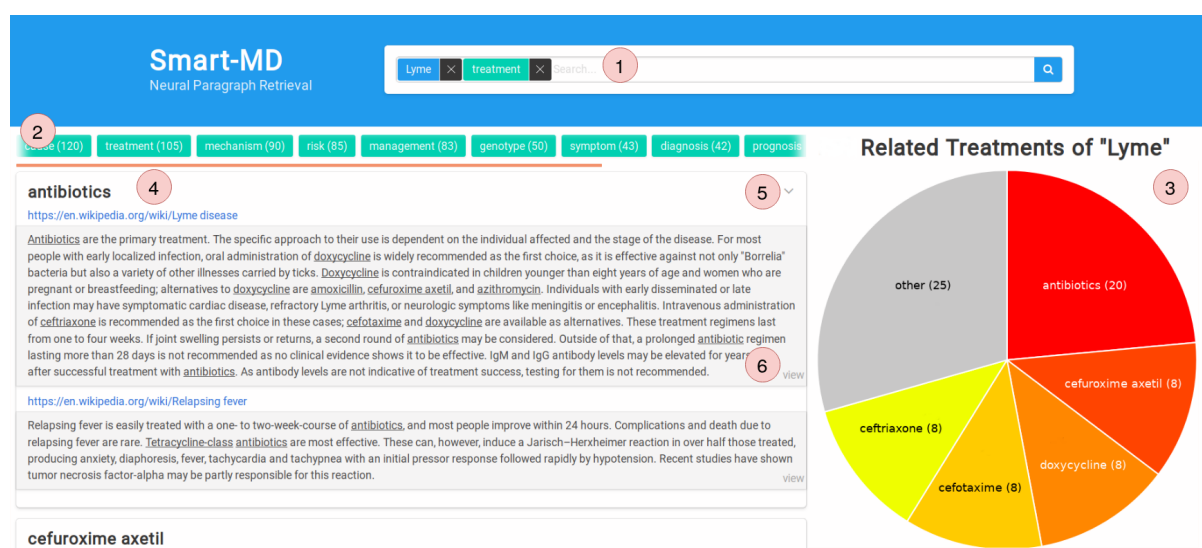


FIGURE 6.6: Screenshot of the SMART-MD user interface.

*entity extractor* recognizes named entities in these sentences. Both models are trained end-to-end with data from the medical domain. We store all extractions in an index and retrieve them at query time to return relevant paragraphs. In this section we describe these steps briefly.

**Sequential topic classification.** The topic extractor's goal is to assign a coherent distribution of topics over all positions in a document. In contrast to traditional probabilistic topic models such as LDA [Blei et al., 2003], which describe topic distributions on document-level, we approach to capture topics on sentence level. To achieve a coherent sequence of topics, e.g. to spot adjacent sentences that express treatments of a disease, we need to respect the sequential order and long-range dependencies of sentences in the document. We use the SECTOR model [Arnold et al., 2019] which utilizes bidirectional Long Short-Term Memory (BLSTM) networks [Hochreiter and Schmidhuber, 1997] to segment and classify passages. The network architecture is shown in Figure 6.7. We utilize section and subsection headlines from Wikipedia documents to define possible topics. For example, we observe 6,876 distinct headlines from 3,469 Wikipedia pages on diseases. A closer inspection reveals that this distribution is heavily skewed, e.g. top 20 topics cover more than 90% of all paragraphs. We therefore chose 20 representative topic labels for training and assign label 'other' to the remainder.

**Medical Named Entity Recognition.** The entity extractor's goal is to recognize medical named entities, such as diseases or medications in the documents. This task is often difficult, since only sparse training data exists and recall suffers from missing variance [Pink et al., 2014]. We utilize the TASTY model [Arnold et al., 2016b], a generic and robust approach for high-recall NER in many languages and with sparse training data. TASTY offers strong generalization over domain-specific language, such as in biomedical text (e.g. Medline, PubMed or Wikipedia articles) and can be trained with only few hundred labeled sentences to achieve  $F_1$  scores in the

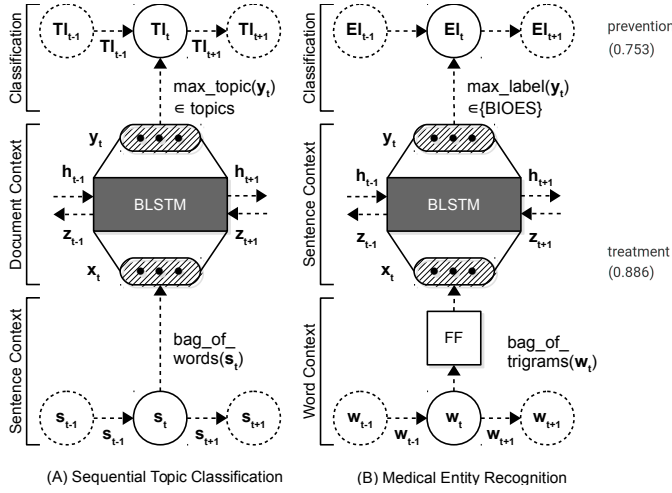


FIGURE 6.7: Neural network architectures for topic classification (left) and entity recognition (right).

Outdoor workers are at risk of Lyme disease if they work at sites with infected ticks. In 2010, the highest number of confirmed Lyme disease cases were reported from New Jersey, Pennsylvania, Wisconsin, New York, Massachusetts, Connecticut, Minnesota, Maryland, Virginia, New Hampshire, Delaware, and Maine. U.S. workers in the northeastern and north-central States are at highest risk of exposure to infected ticks. Ticks may also transmit other tick-borne diseases to workers in these and other regions of the country. Worksites with woods, bushes, high grass, or leaf litter are likely to have more ticks. Outdoor workers should be extra careful to protect themselves in the late spring and summer when young ticks are most active.

Antibiotics are the primary treatment. The specific approach to their use is dependent on the individual affected and the stage of the disease. For most people with early localized infection, oral administration of doxycycline is widely recommended as the first choice, as it is effective against not only *Borrelia* bacteria but also a variety of other illnesses carried by ticks. Doxycycline is contraindicated in children younger than eight years of age and women who are pregnant or breastfeeding; alternatives to doxycycline are amoxicillin, cefuroxime axetil, and azithromycin. Individuals with early disseminated or late infection may have symptomatic cardiac disease, refractory Lyme arthritis, or neurologic symptoms like meningitis or encephalitis. Intravenous administration of ceftriaxone is recommended as the first choice in these cases; cefotaxime and doxycycline are available as alternatives.

FIGURE 6.8: Visualization of neural topic classification on short passages.

range of 84–94% on standard datasets. To achieve a robust classifier, TASTY encodes words as bag of letter-trigrams as input features. This allows us to train a character embedding that is able to recognize typical syllables in a word. We extract possible diseases and other medical entities and store them in the index for query completion and paragraph retrieval.

**Query processing and paragraph scoring.** SMART-MD executes queries of the form (disease, topic) as follows: First, the user matches ambiguous disease and topic names using autocompletion. This operator maps a variety of notations from Wikipedia entity names headlines to well defined entities and topic classes. We then conduct a conjunctive boolean search and retrieve documents that contain both disease name and topic ID a single document. Finally, we score the candidate paragraphs. Our scoring approach bases on the assumption that paragraphs likely contain medical entities that have a mutual relation with the topic of the paragraph and the requested disease. Moreover, we aim to retrieve low frequency events that are probably unknown to the doctor. We measure for each paragraph proximity between the requested topic and co-occurring entities with normalized pointwise mutual information (nPMI) [Bouma, 2009]:

$$\text{nPMI}(\text{entity}, \text{topic}) = \frac{\ln \frac{P(\text{entity}, \text{topic})}{P(\text{entity})P(\text{topic})}}{-\ln P(\text{entity}, \text{topic})} \quad (6.5)$$

$P(\text{entity})$  denotes the probability that retrieved paragraphs contain the entity,  $P(\text{topic})$  the probability that the topic is discussed in the retrieved paragraphs and  $P(\text{entity}, \text{topic})$  denotes the probability that an entity appears in any retrieved paragraph that discusses the topic. Hence we assign to low frequency events relatively high scores and display these results at the beginning of the page.

## 6.4 CDV Healthcare Answer Retrieval

In prior work, we have presented SMART-MD [Schneider et al., 2018], a clinical passage retrieval system based on TASTY Named Entity Extraction [Arnold et al., 2016b] and the SECTOR [Arnold et al., 2019] topic classification method. In this section, extend this task in a system that utilizes the Contextual Discourse Vectors (CDV) model [Arnold et al., 2020] for retrieving clinical answers in long documents. For this scenario, we apply CDV as a Neural Machine Reading model on multiple large corpora of domain-specific text<sup>14</sup>. The aim of this system is to demonstrate a variety of use-cases to doctors and healthcare professionals. From this demonstration, we hope to gain more insights on how we can utilize neural document representations to solve clinical information-seeking tasks.

### 6.4.1 Demonstration Scenario

We use the CDV model, which encodes the discourse of a document, e.g. the entities and aspects discussed in a certain sentence. Our healthcare CDV model is trained with over 27,000 diseases and over 14,000 clinical aspects, such as symptoms, diagnosis, causes, therapy, prevalence, etc. This enables us to search for passages in clinical articles that potentially contain answers for clinical background–foreground questions. We applied CDV to a variety of domain-specific text resources: WikiSection [Arnold et al., 2019], CORD-19 [Wang et al., 2020], Orphanet [INSERM, 1997], MedQuAD [Abacha and Demner-Fushman, 2019] and HealthQA [Zhu et al., 2019], covering over 33.1K articles in total.

Figure 6.9 shows the CDV search interface over the CORD-19 open research dataset<sup>15</sup>. The user can *search* using name of a disease, e.g. “COVID-19”, and the aspect of interest, e.g. “medication” (1). The autocomplete supports her to resolve the correct entities and aspects. The system returns a list of similar entities, e.g. “2019 novel coronavirus respiratory syndrome” and “SARS-CoV-2” which can be clicked to refine the query (2). The search result contains up to 30 passages from different articles that match the query, shown with its matching score in percent (3). By hovering over a sentence, its individual score is shown. For each article, the best matching sentence is highlighted in bold, e.g. “potential drugs [...] such as Remdesivir, Atazanavir, Saquinavir, and Formoterol, and Tocilizumab can be introduced as treatments for COVID-19 [...]”<sup>16</sup>(4). The user can read a larger part of the passage by clicking “more” or open the original source article. By clicking on the headline, the user can access the *highlight* view (5). Here, the entire article is shown, and the correspondence score of each sentence with the query is indicated by the shade of blue. In other words, this view highlights interesting passages that the user should read. This operator is especially helpful for skimming long and complex documents for specific questions.

<sup>14</sup>This system was submitted for review by J.-M. Papaioannou, S. Arnold, F. A. Gers, A. Löser, M. Mayrdorfer, and K. Budde [2020]. “Aspect-Based Passage Retrieval with Contextualized Discourse Vectors”. In: *[IN SUBMISSION] COLING 2020 System Demonstrations*.

<sup>15</sup>A live demo is available at <https://cord19.cdv.demo.dataxis.com>

<sup>16</sup>Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7085862> CC-BY 4.0, 18.04.2020

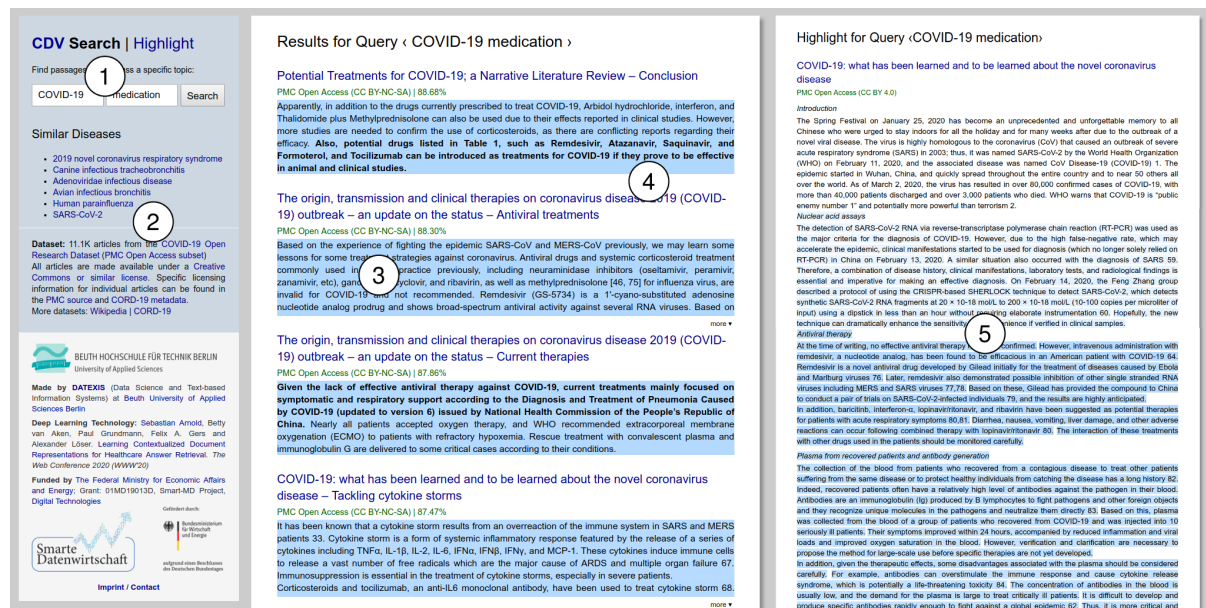


FIGURE 6.9: Screenshot of the CDV search interface (left), result passages (center) and one document in highlight view (right).

Example query	Top 3 sentences	Source
Peanut allergy treatment	"Mild reactions can be treated with an antihistamine medicine."	Patient <sup>17</sup>
	"Antihistamines can alleviate some of the milder symptoms of an allergic reaction, but do not treat all symptoms of anaphylaxis."	Wikipedia <sup>18</sup>
	"The principal treatment for anaphylaxis is epinephrine as an injection."	Wikipedia <sup>19</sup>
Cystic fibrosis symptoms	"The main signs and symptoms of cystic fibrosis are salty-tasting skin, poor growth, and poor weight gain despite normal food intake, accumulation of thick, sticky mucus, frequent chest infections, and coughing or shortness of breath."	Wikipedia <sup>20</sup>
	"Signs and symptoms may include salty-tasting skin; persistent coughing; frequent lung infections; wheezing or shortness of breath; poor growth; weight loss; greasy, bulky stools; difficulty with bowel movements; and in males, infertility"	GARD <sup>21</sup>
	"The most common form of cystic fibrosis is associated with respiratory symptoms, digestive problems [...] and staturponderal growth anomalies."	Orphanet <sup>22</sup>
COVID-19 medication	"Also, potential drugs [...] such as Remdesivir, Atazanavir, Saquinavir, and Formoterol, and Tocilizumab can be introduced as treatments for COVID-19 if they prove to be effective in animal and clinical studies"	PMC <sup>16</sup>
	"Chloroquine has been used to treat malaria for many years, with a mechanism that is not well understood against some viral infections."	PMC <sup>23</sup>
	"The WHO does not oppose the use of non-steroidal anti-inflammatory drugs (NSAIDs) such as ibuprofen for symptoms, and the FDA says currently there is no evidence that NSAIDs worsen COVID-19 symptoms"	Wikipedia <sup>24</sup>

TABLE 6.2: Example results for three CDV healthcare queries. The table shows for each query the highlighted sentences from the top-3 predicted passages.

<sup>17</sup>Source: <https://patient.info/allergies-blood-immune/food-allergy-and-intolerance/nut-allergy>, 18.04.2020

<sup>18</sup>Source: [https://en.wikipedia.org/wiki/Food\\_allergy](https://en.wikipedia.org/wiki/Food_allergy) CC-BY-SA 3.0, 18.04.2020

<sup>19</sup>Source: [https://en.wikipedia.org/wiki/Peanut\\_allergy](https://en.wikipedia.org/wiki/Peanut_allergy) CC-BY-SA 3.0, 18.04.2020

<sup>20</sup>Source: [https://en.wikipedia.org/wiki/Cystic\\_fibrosis](https://en.wikipedia.org/wiki/Cystic_fibrosis) CC-BY-SA 3.0, 18.04.2020

### 6.4.2 Discussion of Healthcare Queries

We exemplify the results for three healthcare queries on six different datasets shown in Table 6.2. We chose these queries because they show a variety of *exploratory* information-seeking tasks, such as *lookup*, *learn* and *investigate* [Marchionini, 2006].

“Treatments for peanut allergy” is a typical consumer question, which is difficult to answer with a single *fact lookup*, because there exists no cure for this allergy up to now. Such queries require the user to *learn* more about a topic, so they are best answered by passages from health portals like Patient or the Wikipedia encyclopedia. We notice that two articles mention antihistamines for mild reactions, and another article suggests epinephrine (a synthetic form of adrenaline) as a treatment for anaphylaxis. The information that anaphylaxis is a severe allergic reaction is mentioned in the context of this passage. We further notice that the information is contained in specific articles about peanut allergy, but also in more generic ones such as food allergies. Therefore it is important that the predictions respect the locality of the passages.

“Symptoms of Cystic fibrosis” is a similar *lookup* question that is focused on a *rare disease*. Here, the answers are contained in various specialized sources such as GARD or Orphanet, which contain only short articles. We observe that the three top answers strongly overlap, some describe the symptoms more casually (e.g. poor growth), others are more specific (e.g. staturponderal growth anomalies). We further notice that for these rare cases, answers are very precise and there is also string lexical overlap between query and passage.

“Medication for COVID-19” is an *investigatory* question which focuses on finding potential medications for the new COVID-19 disease. This is an open-ended question, and we cannot validate its answers today. Therefore, the goal is to maximize the recall over recent research articles and point an expert user to the interesting passages. The answers show a high variance and discuss potential drugs such as Remdesivir, Formoterol or Chloroquine, which are discussed in individual PMC articles. The model also predicts a more generic Wikipedia passage mentioning ibuprofen as a typical anti-inflammatory drug that has been widely discussed in relation with COVID-19.

<sup>21</sup>Source: <https://rarediseases.info.nih.gov/diseases/6233/index> 18.04.2020

<sup>22</sup>Source: [https://www.orpha.net/consor/cgi-bin/Disease\\_Search\\_Simple.php?lng=EN&diseaseGroup=Cystic+fibrosis](https://www.orpha.net/consor/cgi-bin/Disease_Search_Simple.php?lng=EN&diseaseGroup=Cystic+fibrosis) Copyright INSERM 1997, 18.04.2020

<sup>23</sup>Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7068984> CC-BY 4.0, 18.04.2020

<sup>24</sup>Source: [https://en.wikipedia.org/wiki/Coronavirus\\_disease\\_2019](https://en.wikipedia.org/wiki/Coronavirus_disease_2019) CC-BY-SA 3.0, 18.04.2020



## 6.5 Conclusions

In this chapter, we have introduced four systems that cover a broad range of Machine Reading applications. TASTY combines classical Information Extraction with an interactive feedback loop while the user is writing text. TraiNER approaches the efficient creation of domain-specific Information Extraction models in an active learning setting. SMART-MD models a clinical information-seeking task with a combination of neural Information Extraction and Machine Reading methods over long documents. Eventually, CDV Healthcare Retrieval utilizes a self-supervised Neural Machine Reading model to retrieve answers from a variety of domain-specific text resources. With these systems, we cover the entire process of supporting human information seeking as introduced in Section 1.1. Most of these systems are prototypical implementations that have been applied in industry and research projects. These systems benefit from the contributions in scope of this thesis, which are focused on automatic language processing. It remains for future work to evaluate the entire information-seeking process—which includes the user interfaces and feedback processes of these systems—with methodologies from human-computer interaction (HCI).



## Chapter 7

# Conclusion and Future Work

In this thesis, we have examined the vision of Neural Machine Reading for domain-specific text resources from a variety of viewpoints. We have contributed the Neural MR architectures TASTY, SECTOR and CDV that support three central tasks in the process of human information seeking: Named Entity Linking, Topic Modeling and Answer Passage Retrieval. We have shown that deep neural networks enable the efficient creation of generalized models using end-to-end training methods, self-supervision and the integration of contextual and background knowledge. Our models are able to process domain-specific text resources without expensive adaptation and with high error tolerance. We have evaluated several information-seeking tasks and shown that our models achieve high accuracy, while they are often trained with widely available training data, e.g. from Wikipedia. In this chapter, we review our contributions with respect to the desired properties of a MR model (Section 7.1) and research questions (Section 7.2). We discuss the limitations (Section 7.3) and perspectives (Section 7.4) of our vision. Finally, we propose directions for future work (Section 7.5).

## 7.1 Contributions of Neural Machine Reading

In Section 1.1, we have introduced six central challenges for domain-specific language understanding. We have designed our models to meet these requirements and now discuss our contributions and findings:

**Domain-specific language understanding.** We have approached three central tasks over multiple languages or domains. More specifically, we proposed a deep learning architecture for Named Entity Recognition that can be trained end-to-end and does not require language and domain-specific feature engineering (Section 3.2.2). We have shown that our NER model can efficiently adapt to English and German in news, biomedical and industry domains, when supervised training data is available (Section 3.3). We further have introduced the SECTOR and CDV Topic Classification and Answer Passage Retrieval architectures that share these properties, even when trained with self-supervision from Wikipedia section headings (Sections 4.3.2 and 5.3). We have shown that SECTOR adapts to English and German for medical, geopolitical, chemistry and clinical domains with high accuracy (Section 4.4). We further report high

accuracy of CDV for retrieving answers from medical encyclopedia, consumer healthcare, clinical research and professional biomedicine domains, even without any additional fine-tuning (Sections 5.4 and 6.4).

**Robustness against noise and spelling variations.** We have analyzed errors arising from capitalization, spelling variations, novel or incomplete words and irregular sentences (Section 3.1.3). All of our Machine Reading approaches abstract from rule-based linguistic preprocessing in order to avoid these errors. Instead, they rely on the distributional hypothesis to represent language using empirical observations, such as distributed word embeddings and language models (Section 2.2). Primarily, we have shown that letter-trigram encoding is a key component for robust word representations (Section 3.2.1). We further have shown that sentence embeddings based on Bloom filters, weighted average or self-attention over words improve recall for information-seeking tasks (Sections 4.3.1 and 5.3.1).

**Document structure representation.** We have shown that our SECTOR and CDV models are able to encode the structural properties of long documents (Sections 4.3 and 5.3). Our document representations improve information-seeking tasks with contextualized local information that is not contained in word or sentence based models, such as Word2Vec, ParVec or BERT (Sections 4.5, 5.3.3 and 6.4).

**Broad task coverage.** We have covered a broad range of tasks using two approaches. First, our efficient supervised training techniques for TASTY and TraiNER allow to build personalized IE models from labeled data without any architecture changes (Sections 3.2.2 and 6.2). We have shown high accuracy on Named Entity Recognition and Linking to Wikipedia, but also report high recall from extracting general concepts, noun phrases, more specific biomedical terms, car models and car parts (Section 3.3). However, these models rely on accurate training data and are not easily interchangeable between tasks. Therefore our second approach with CDV focuses on general Machine Reading models that can be reused for different tasks. We have shown that distributed entity and aspect embeddings are able to handle generic and specific Answer Passage Retrieval tasks using nearest-neighbor search (Section 5.2). This includes for example general disease descriptions, precise definitions of rare diseases and also zero-shot adaptation to previously unknown diseases such as COVID-19 (Section 6.4). We further have shown that multi-task training with complementary objectives, such as named entities and topical aspects, improve representations for Answer Retrieval compared to language models that solely rely on the distributional hypothesis (Section 5.4).

**Efficient model training.** We have proposed several approaches to deal with insufficient amounts of training data. Our TASTY model utilizes efficient word representations, sequential context and background knowledge to train NER models with high accuracy using only 4,000-5,000 labeled sentences (Section 3.3). We have proposed the TraiNER framework to reduce the amount

of human labeling for this task to few hundred examples by automatically matching seed labels and sampling instances for active learning (Section 6.2). We proposed SECTOR and CDV document representations that can be reused for downstream tasks. These models are efficiently trained with self-supervised data from Wikipedia articles and leverage background knowledge from large pre-trained language models and complementary entity and aspect embeddings (Sections 4.2 and 5.3.4).

**Error analysis and feedback propagation.** We have conducted error analyses for each of our models to explain their strengths and weaknesses (Sections 3.4, 4.5 and 5.5). We aimed to cover the entire process of human information seeking by including the user into the feedback loop of the TASTY Editor and TraiNER active learning system. The components of our models are trained end-to-end, so they can be replaced with updated weights and often improve with more feedback (Sections 6.1 and 6.2).

## 7.2 Review of Research Questions

In the beginning of this thesis, we posed four central research questions to approach our hypothesis of Neural Machine Reading. We are now going to summarize our findings for each of these questions.

**RQ1. *What are general solutions to identify named entities in domain-specific text?*** Extracting named entities from domain-specific text requires a model that can leverage local, contextual and global features. First, we identified character-based word representations as a key component for efficient and robust recognition of domain-specific entity names. In particular, letter-trigram encodings provide our model with important local subword information that enables the model to efficiently learn from sparse data with high recall. Second, we have shown that Bidirectional Long Short-Term Memory (BLSTM) models effectively encode long-range dependencies from sentence and document context. The encoder-decoder architecture using stacked BLSTMs enables us to train language-invariant models for Named Entity Recognition end-to-end, i.e. using a labeled set of 4,000–5,000 examples or in an active learning scenario. Third, pre-trained language models, word and sentence embeddings provide important background information for generalization of the model. We used a combination of these features to achieve 91.1%  $F_1$  on the English CoNLL03 task, and report high scores for domain-specific NER models in English and German. Furthermore, we used entity embeddings that were trained with plain entity descriptions to efficiently retrieve and rank candidates for Named Entity Disambiguation and Answer Passage Retrieval with high accuracy using nearest neighbor search.

**RQ2. *How can Machine Reading models detect topics and structure in long documents?*** The understanding of entire documents is an important ingredient for Machine Reading. To tackle

this task, we introduced a topic segmentation and classification task which operates with sentence granularity on long documents with 1,500 words on average. We have shown that models based on the distributional hypothesis, such as Paragraph Vectors or LDA topic models, can not solve this task adequately. Instead, MR models require complementary structural information, which we take from section headings of Wikipedia training documents. We have shown that BLSTMs can effectively capture entire documents using Bloom filters for sentence encoding. We have introduced a bidirectional embedding deviation method, inspired by edge detection in images, to segment documents at topic shifts into coherent passages with high accuracy. We further have shown that the same MR model can be used to classify passages into 25–30 normalized topic classes with up to 71.6%  $F_1$ . We provided insights showing that the model predicts a coherent topical structure, which can further be reused for downstream tasks such as large multi-class multi-label classification with up to 603 classes.

**RQ3. *How can we embed discourse structure into document representations?*** Automatic language understanding requires a contextualized document representation that reflects the discourse structure of a text, including topical structure, entity mentions, coreferences and general long-range dependencies. We proposed to extend pre-trained language models that rely on the distributional hypothesis with complementary information from document structure and distributed entity and aspect embeddings. We integrated these objectives using multi-task training over entire documents with sentence granularity. We have introduced a model that uses BLSTMs stacked on top of distributed sentence representations and Huber loss to align the sentences of a document with contextual discourse information. This step was possible without external supervision, because Wikipedia documents provide enough features for the alignment. Furthermore, we showed that by integrating a pre-trained BERT language model as lowest layer, accuracy improves significantly. We have discussed that our document representation retains the original properties of the semantic entity and aspect spaces, such as nearness measures, robustness against variations and the coverage of long-tail entities. This further helped the model to generalize over previously unseen examples and provides semantic interpretability of the representation vector space.

**RQ4. *How effective are document representations for retrieving answer passages?*** We examined the application of our contextual discourse vector representation in an answer retrieval task. We have shown that searching contextualized document representations on sentence level using cosine similarity achieves significantly higher recall than term-based methods and shows equal to superior performance compared with supervised document re-ranking methods. This is possible because structural document context and long-range dependencies are normally not captured by document matching models. Furthermore, we highlighted that representation-based search is more efficient than models based on deep interaction between query and document. Document representations can be precomputed and cached, so that the complexity of an ad-hoc query is reduced to encoding the query and retrieving k-nearest

neighbors from a vector space index. From an in-depth error analysis, we identified the representation of hierarchical, related and overlapping information as a potential cause of errors, because this information is not considered adequately by the cosine similarity measure.

Put together, our answers to these research questions enable us to build general Neural Machine Reading models that fulfill task-specific information needs across domain-specific text resources. This thesis covers all necessary stages of this process. We have built our contribution around the definition of central information-seeking tasks, the principles of unsupervised language understanding, distributed language representations and sequence learning methods. We have contributed algorithms that solve three central tasks with high accuracy and high error tolerance from self-supervised data or only few hundred labeled examples. Even though many of the individual problems have been approached by fast-paced concurrent work, no comprehensive solution to document-level Neural Machine Reading has been presented before. This thesis is the first research approach to extend distributed language representations with complementary information about document topics and discourse structure. It closes the gap between symbolic Information Extraction and Information Retrieval by transforming both problems into latent distributed vector space representations. Our models can fulfill domain-specific information needs on large domain-specific text resources with low latency suitable for interactive applications.

### 7.3 Limitations

We have presented a general architecture for Neural Machine Reading that is applicable to a broad range of domain-specific text resources. As every research project, our approach is subject to a number of limitations, which could be addressed in future work.

**Applicability to different writing systems.** We could show that end-to-end models work well when transferred to different languages, such as English and German. However, our experiments have been restricted to written languages that use a linear segmented monophonemic alphabet, such as Latin script. This is mainly caused by common preprocessing steps, such as tokenization, sentence splitting and character encoding, which rely on the assumption of space-separated character n-grams. These operations are often rule-based and language-specific, although recent NLP libraries cover a large variety of languages. We further cannot make any assumptions that the principles of the distributional hypothesis will also hold in languages that use morphemic (e.g. Chinese) or partial phonemic (e.g. Arabic) writing systems. One possible circumvention would be to replace preprocessing steps with compatible methods, such as Chinese word segmentation algorithms [Peng et al., 2004]. This could enable us to train each individual component of the MR model with end-to-end training data of the different writing system.

**Preprocessing required for self-supervision.** The approaches for self-supervised training introduced in this thesis are not entirely unsupervised. Generating self-supervised training data from external sources such as Wikipedia often requires a fair amount of site-specific structural parsing, e.g. extracting document titles, links, lists and section headings from the HTML source. This is not always possible on domain-specific text resources, because they might not expose this structural information. To better understand this limitation, we have investigated the transfer of models trained from Wikipedia to domain-specific text and demonstrated options to minimize the annotations required for fine-tuning. One possible circumvention is to specifically train models that select an ensemble of hand-written heuristics in order to generate training data from unlabeled corpora with weak supervision [Ratner et al., 2020].

**Controlled adjustment of model properties.** We could show that our Neural Machine Reading architectures solve three important information-seeking tasks with high accuracy compared to a variety of recent approaches. However, in practice, for example in Web search engines, accuracy measures such as Precision and Recall are not the only objectives. Instead, optimization criteria for search engine result pages are domain-specific and dynamically change between user profiles and from user feedback. Typical objectives include result diversification, freshness, source trust, popularity, etc. [Toms et al., 2005; Chuklin et al., 2013; Li et al., 2015]. In Information Retrieval, these objectives are often achieved by learning personalized decision rules between an ensemble of multiple models by relevance feedback. Additionally, often experts curate hard-coded rules and exceptions to maximize the key performance indicators of their product. An ideal solution would be to enable these adjustments and feedback inside the model itself, or by extending the model with controlled ‘plug&play’ layers [Dathathri et al., 2020]. Although we have shown that our neural MR architecture is highly adaptive to a broad range of tasks and domains, we have not focused on dynamically changing objectives and leave a solution up to future work.

## 7.4 Business Perspectives

Neural Machine Reading opens up a broad range of perspectives for commercial applications. Most importantly, general and robust MR models accelerate the design of *data products* which normally require time-intensive research and development. These products are urgently needed to conquer the continuously growing data lake from corporate information resources, social digital communication and the Web in general. It is assumed that by 2025, 80% of worldwide data will be unstructured, with healthcare, manufacturing, financial services and retail being the fastest-growing industries [Reinsel et al., 2018]. Currently, only a small fraction of this data is analyzed, although approximately one quarter could contain valuable information [Gantz and Reinsel, 2012]. For example, according to a 2019 study, only 11% of manufacturing companies consider themselves as ‘data mature’ and only 48% have begun the transformation of the organization towards data-driven processes [Atkinson and Ezell, 2019]. The most

prominent barriers for this transformation mentioned in this study are lack of data resources (58%), task-specific implementation challenges (52%), lack of development skill (47%) and interoperability and integration problems (47%). A general-purpose MR model addresses these problems.

**Increasing coverage of domain-specific text resources in e-discovery.** Electronic discovery (e-discovery) is a legal process in which a party requests the delivery of electronically stored information (ESI) as potentially relevant evidence in a civil lawsuit. Typical search queries include the application of patents, misuse of licensing rights, or collusion. A key challenge to the e-discovery process is to cover a broad range of resources (e.g. corporate documents, e-mail, instant messaging communication, audio recordings, databases, Web sites, images, metadata etc.), to deliver only specific information that is relevant for litigation and, at the same time, to protect sensitive corporate data.

With growing emergence of new content sources in organizations, rising number of litigations, increasing growth in compliance requirements and data protection regulations, e-discovery is expected to gain major traction in the next years. According to a recent study, the global e-discovery market is growing at a compound annual growth rate of 10.0% and is projected to reach over \$17.3 billion by 2023 [MarketsandMarkets, 2019]. Discovery accounts for 20–50% of all costs in federal civil litigations [Lee and Willging, 2010]. Therefore, organizations need to proactively prepare for delivery requests from potential lawsuits in order to minimize the cost for manual management. For example, in a past patent dispute of Samsung against Apple Computers, 11 million documents of over 3.6 terabytes were processed, with a total processing cost of over \$13 million [Sullivan, 2017]. Today, e-discovery is built upon the *Electronic Discovery Reference Model* (EDRM), a standard process for information governance, identification, preservation, collection, processing, review, analysis, production and presentation of ESI [Holley et al., 2010]. The main contributions of e-discovery software is to reduce complexity of the process and strip noise from the data. However, text resources are not deeply processed in these systems, and in particular review and analysis stages still require expensive human labor. Typical solutions, such as Microsoft Office 365 eDiscovery<sup>1</sup>, IBM StoredIQ Suite<sup>2</sup>, DISCO<sup>3</sup> or Logikcull<sup>4</sup> are based on term-based keyword search and metadata filters. Some of these systems apply machine learning approaches for keyword and topic expansion.

Neural Machine Reading supports the e-discovery process fundamentally in the important stages of identification, collection, review and analysis of textual data. It helps to accelerate these stages in order to save costs by providing general language representations, which can be trained end-to-end with self-supervised data and can be queried with low latency. Neural MR representations provide a higher result coverage by effectively increasing recall over large corpora of domain-specific resources, in particular from the long tail. Entity and topical aspect

<sup>1</sup><https://docs.microsoft.com/en-us/microsoft-365/compliance/ediscovery>

<sup>2</sup><https://www.ibm.com/products/storediq-suite>

<sup>3</sup><https://www.csdisco.com/disco-ediscovery>

<sup>4</sup><https://www.logikcull.com/>

representations condense important latent concepts for searching, clustering and visualizing data. Document representations can be used to retrieve short passages from long documents. In summary, Neural MR provides the e-discovery process with in-depth semantic coverage of text resources and guides experts to make review decisions faster and with higher precision.

**Detecting trigger events for Supply-Chain Risk Management.** Supply-chain risk management (SCRM) describes the strategies to identify, assess, control and monitor unforeseen developments and their effects on a supply chain [Heckmann et al., 2015]. A key process in SCRM is to identify the occurrence of *triggering events* for a long list of risk factors, such as geopolitical and economic instability, environmental risks, weather events, natural disasters, technical failures, crime, transportation issues, product issues, market uncertainty, corporate transactions, stock market activity and legal issues. For example, pharmaceutical companies must secure multiple weeks of supply for critical drugs and emergency equipment. The manufacturing industry is characterized by just-in-time production and needs to minimize the potential impact of logistic delays and disruptions. Financial services and insurance companies need to assess global economic and environmental risk factors and proactively detect stock market and economic disruptions with short reaction times. For these organizations, it is viable to identify public mentions of these events at their first occurrence. This is possible by screening a broad range of internal and external information resources. The key challenges in event detection are to deliver results with low latency, to achieve high recall without triggering false alarms, and to reduce the number of cases that require manual examination.

In a 2019 survey among procurement leaders, managing risk was the second most important business strategy (55% strong priority), right after reducing costs (70%) [Umberhauer et al., 2019]. Moreover, 81% of companies in the same survey who have fully implemented digital technologies, report that they are not satisfied with their SCRM implementations. More specifically, 65% of procurement leaders have reported limited or no transparency in the supply chain beyond their first-level suppliers [Umberhauer and Younger, 2018]. Installing automated information-driven processes for SCRM can reduce costs for procurement, inventory management and logistics by identifying risks with short reaction times, assessing risks with high precision and minimizing risks by proactive actions. Today, intelligent analysis systems with global coverage deliver frequent and recurrent data from news outlets, social media, company websites, press releases, stock exchange reports, audit reports or financial reports. The main functions of these systems comprise automatic tagging, de-duplication scoring of risk trigger events. However, these systems are mainly focused on procurement and logistics sectors and do not deliver sufficient quality of results from more specific domains. Especially small tech companies and startups operate in highly specialized niches and need to detect rare events with high recall.

Neural Machine Reading provides the necessary technology to complement SCRM solutions by enriching and annotating text resources with domain-specific entities, aspects, topics and events. By automatic language understanding, the high complexity of dependencies is



reduced to semantic representations that can be utilized by a company to define its specific risk areas with high granularity. Neural MR is specifically important for handling the cases of zero-shot adaptation to unknown sources with sparse and noisy data. Furthermore, distributed representations allow to visualize results and accelerate human examination and labeling. In summary, Neural Machine Reading improves SCRM with higher coverage of risk trigger events, lower costs for manual examination and better semantic accessibility for decision makers and creators of risk assessment reports.

**Enriching patient representations in electronic healthcare.** Clinical Decision Support Systems (CDSS) aim to capture a comprehensive picture of a patient in order to assist clinicians in their choices at the point of care [Berner, 2007]. A key process in CDSS is the integration of multi-modal electronically stored information, such as laboratory test results, vital signs, radiology images and clinical notes into electronic health records (EHR). However, a large fraction of EHRs consists of written notes. These are regularly created and updated by medical professionals and comprehend the entire trajectory of a patient, including admission (chief complaints, results of physical examination), medical history (surgeries, medications, family and social history), therapy progress, test results and discharge information (diagnosis, medications, prognosis). This longitudinal clinical pathway information provides important information for differential diagnosis (DDx) and supports doctors to decide diagnostic procedures or treatments. Electronic healthcare faces the problem to make these notes machine readable and allow a CDSS to access rich patient representations. Patient representations allow searching, clustering or even simulating patient trajectories based on similar cohorts from past medical records. This would enable clinicians to make decisions earlier and with the reduced risk of overlooking important cases from the large historic record.

Healthcare annual data production is projected to grow by 36% between 2018–2025 [Reinsel et al., 2018]. The market for CDSS is growing at a rate of 9.5% and is projected to reach \$2.4 billion by 2027 [Reports and Data, 2020]. Improving patient representations could accelerate the diagnosis of rare diseases, reduce length-of-stay, propose cheaper medication options, decrease test duplication and provide better informed reasoning to the clinicians. Current CDSSs resemble clinical guidelines and are mainly focused on capturing structured data and metadata, such as symptoms and conditions extracted from clinical text, and visualize these findings to allow exploration of cases. Typically, these expert systems allow term-based and metadata search to explore literature and studies or identify cohorts from case reports. However, even large hospitals fail to integrate their own EHRs with these systems. This is partly caused by technological and data protection boundaries, but also because off-the-shelf text mining models fail to adapt to doctor’s specific writing styles.

Neural Machine Reading provides an opportunity to deeply integrate clinical notes with EHRs. Neural patient representations can be inferred from textual data, such as doctor letters and admission notes and cover a broad range of clinical aspects that are otherwise not contained in structured EHR data and metadata. Vector space representations provide similarity

measures that enable precise search for cohorts, similar cases from clinical guidelines and case studies, and research literature. Neural MR models can be trained on public data, fine-tuned with small amounts of anonymized in-domain data, and applied to sensitive personal data in-house. Vector space indexing can help to precompute and access passages from thousands of clinical EHRs with low latency. In summary, Neural Machine Reading improves patient representations with deep understanding of domain-specific clinical notes and makes it possible to search, cluster and explore EHRs directly based on the information provided by doctors and clinicians.

## 7.5 Future Work

During our research on Neural Machine Reading for domain-specific text resources, we identified several meta-problems that we had to leave for further investigation. In this chapter, we briefly discuss the most important questions as a guideline for future work.

**Hierarchical knowledge representations.** Generalization and specialization are important concepts in human cognition. Therefore, *hierarchical knowledge* is often modeled explicitly in information management [Saxe et al., 2013]. For example, medical reference knowledge bases, product catalogs or geographical databases are structured into broad categories and multiple levels of finer-grained subcategories. Entries in these data structures are often located in all hierarchy levels, sometimes including associations between multiple overlapping categories. We have shown that the vector space model applied with cosine similarity is error-prone, especially when associations between different nested or overlapping hierarchical levels have to be modeled. Furthermore, vector space queries do not allow basic query composition operators, such as union, intersection and complement. For example, in our experiments queries for very specific cancer types often showed high similarity with generic descriptions of cancer and tumors, and we needed to sharpen or broaden the range of a query. More precisely, the distributional hypothesis succeeds at drawing associations between multiple elements in a nested hierarchy, but it fails to distinguish them and exploit their hierarchical position. One approach for this problem are hierarchical entity embeddings [Hu et al., 2015]. Another promising method is to compute embeddings not in Euclidean, but in hyperbolic space, such as the Poincaré ball model [Nickel and Kiela, 2017].

**Understanding real-world data by modeling uncertainty.** When applied to real-world problems, probabilistic models will always produce errors based on the difference of data distributions at training and inference time. However, current DNN architectures are not able to report their confidence on a prediction. Extending our neural MR model with *uncertainty modeling* would enable us to sample and counteract weak points of our models more effectively, for example using active learning. It should be our objective to reduce high-confidence false

predictions and reinforce low-confidence true predictions of a model. Furthermore, measuring different aspects of uncertainty is an important requirement for safety-critical operations [Kendall and Gal, 2017]. One promising approach for this problem is *Bayesian Deep Learning*, which uses the principles of neural variational inference [Paisley et al., 2012; Ranganath et al., 2014] and can be estimated using existing DNN architectures with Monte Carlo dropout sampling [Gal and Ghahramani, 2016].

**Continual learning from interactive feedback loops.** Whenever new data is produced, predictions will deviate from the expected results that the model was optimized for at training time. To counteract this problem, neural MR models need to be retrained from time to time with fresh in-domain data or new tasks. This is possible by introducing new prior distributions from self-supervised examples, or by collecting explicit or implicit feedback from users in an active or passive learning scenario. Sometimes, it is even feasible to make controlled adjustments to a model in order to reduce or introduce certain biases, e.g. balancing recall and precision for business-critical classification outputs. This *continual learning* process introduces a number of challenges [Parisi et al., 2019; Yogatama et al., 2019]: How can we ensure enough capacity in a network for lifelong learning? What is the optimal curriculum for updating models with new tasks sequentially? How can we prevent *catastrophic forgetting* of previously learned tasks? A promising approach for this problem is to extend transfer learning with episodic memory modules that allow *experience replay* of previous examples during training [de Masson d’Autume et al., 2019].

Put together, the research directions of hierarchical knowledge representations, uncertainty modeling and continual learning aim towards extending Neural Machine Reading with three important properties: MR models need to acquire a deeper understanding about the domain’s structural and hierarchical properties. We need to better understand the models’ inner workings, uncertainties and limitations. And we require feedback operators to handle dynamic changes of the environment where the models are applied. With this perspective, Machine Reading models will be able to complement domain-specific language understanding with automatic task understanding based on users’ information-seeking behavior.



# Bibliography

- Abacha, A. B. and Demner-Fushman, D. (2019). "A Question-Entailment Approach to Question Answering". In: *BMC Bioinformatics* 20.1, p. 511.
- Abacha, A. B., Shivade, C., and Demner-Fushman, D. (2019). "Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering". In: *Proceedings of the 18th BioNLP Workshop and Shared Task*, pp. 370–379.
- Adolphs, P., Theobald, M., Schäfer, U., Uszkoreit, H., and Weikum, G. (2011). "YAGO-QA: Answering Questions by Structured Knowledge Queries". In: *Fifth International Conference on Semantic Computing*. IEEE, pp. 158–161.
- Agarwal, S. and Yu, H. (2009). "Automatically Classifying Sentences in Full-Text Biomedical Articles into Introduction, Methods, Results and Discussion". In: *Bioinformatics* 25.23, pp. 3174–3180.
- Ajjour, Y., Chen, W.-F., Kiesel, J., Wachsmuth, H., and Stein, B. (2017). "Unit Segmentation of Argumentative Texts". In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 118–128.
- Akbik, A., Blythe, D., and Vollgraf, R. (2018). "Contextual String Embeddings for Sequence Labeling". In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649.
- Alemi, A. A. and Ginsparg, P. (2015). "Text Segmentation Based on Semantic Word Embeddings". In: *arXiv:1503.05543 [cs.CL]*.
- Allan, J. (2002). "Introduction to Topic Detection and Tracking". In: *Topic Detection and Tracking*. Springer, pp. 1–16.
- AlSumait, L., Barbará, D., and Domeniconi, C. (2008). "On-Line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking". In: *Eighth IEEE International Conference on Data Mining*. IEEE, pp. 3–12.
- Aone, C., Halverson, L., Hampton, T., and Ramos-Santacruz, M. (1998). "SRA: Description of the IE2 System Used for MUC-7". In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.
- Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., Kameyama, M., Martin, D., Myers, K., and Tyson, M. (1995). "SRI International FASTUS System: MUC-6 Test Results and Analysis". In: *Proceedings of the 6th Conference on Message Understanding*. ACL, pp. 237–248.
- Arnold, S., Burke, D., Dörsch, T., Loeber, B., and Lommatzsch, A. (2014). "News Visualization Based on Semantic Knowledge." In: *International Semantic Web Conference (Posters & Demos)*, pp. 5–8.

- Arnold, S., Dziuba, R., and Löser, A. (2016a). "TASTY: Interactive Entity Linking As-You-Type". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 111–115.
- Arnold, S., Gers, F. A., Kiliyas, T., and Löser, A. (2016b). "Robust Named Entity Recognition in Idiosyncratic Domains". In: *arXiv:1608.06757 [cs.CL]*.
- Arnold, S., Löser, A., and Kiliyas, T. (2015). "Resolving Common Analytical Tasks in Text Databases". In: *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP (DOLAP)*. ACM, pp. 75–84.
- Arnold, S., Schneider, R., Cudré-Mauroux, P., Gers, F. A., and Löser, A. (2019). "SECTOR: A Neural Model for Coherent Topic Segmentation and Classification". In: *Transactions of the Association for Computational Linguistics* 7, pp. 169–184.
- Arnold, S., van Aken, B., Grundmann, P., Gers, F. A., and Löser, A. (2020). "Learning Contextualized Document Representations for Healthcare Answer Retrieval". In: *Proceedings of The Web Conference 2020*. ACM, pp. 1332–1343.
- Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). "A Latent Variable Model Approach to PMI-Based Word Embeddings". In: *Transactions of the Association for Computational Linguistics* 4, pp. 385–399.
- Arora, S., Liang, Y., and Ma, T. (2017). "A Simple but Tough-to-Beat Baseline for Sentence Embeddings". In: *ICLR 2017: 5th International Conference on Learning Representations*.
- Atkinson, R. D. and Ezell, S. J. (2019). *The Manufacturing Evolution: How AI Will Transform Manufacturing & the Workforce of the Future*. Tech. rep. Manufacturers Alliance for Productivity and Innovation (MAPI) and Information Technology and Innovation Foundation (ITIF).
- Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). "The Berkeley Framenet Project". In: *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*. ACL, pp. 86–90.
- Barron, J. T. (2019). "A General and Adaptive Robust Loss Function". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339.
- Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". In: *The Annals of Mathematical Statistics* 41.1, pp. 164–171.
- Bayomi, M., Levacher, K., Ghorab, M. R., and Lawless, S. (2015). "OntoSeg: A Novel Approach to Text Segmentation Using Ontological Similarity". In: *2015 International Conference on Data Mining Workshop*. IEEE, pp. 1274–1283.
- Beam, A. L., Kompa, B., Schmaltz, A., Fried, I., Weber, G., Palmer, N., Shi, X., Cai, T., and Kohane, I. S. (2018). "Clinical Concept Embeddings Learned from Massive Sources of Multimodal Medical Data". In: *Pacific Symposium on Biocomputing*. Vol. 25, pp. 295–306.
- Beeferman, D., Berger, A., and Lafferty, J. (1999). "Statistical Models for Text Segmentation". In: *Machine Learning* 34.1, pp. 177–210.

- Bender, O., Och, F. J., and Ney, H. (2003). "Maximum Entropy Models for Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4*. ACL, pp. 148–151.
- Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). "A Neural Probabilistic Language Model". In: *Journal of machine learning research* 3.Feb, pp. 1137–1155.
- Berner, E. S. (2007). *Clinical Decision Support Systems*. Vol. 233. Springer.
- Bhatia, S., Lau, J. H., and Baldwin, T. (2016). "Automatic Labelling of Topics with Neural Embeddings". In: *Proceedings of the 26th International Conference on Computational Linguistics*, pp. 953–963.
- Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). "Nymble: A High-Performance Learning Name-Finder". In: *Proceedings of the Fifth Conference on Applied Natural Language Processing*. ACL, pp. 194–201.
- Black, W. J., Rinaldi, F., and Mowatt, D. (1998). "FACILE: Description of the NE System Used for MUC-7". In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.
- Blei, D. M. (2012). "Probabilistic Topic Models". In: *Communications of the ACM* 55.4, pp. 77–84.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3.Jan, pp. 993–1022.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146.
- Bouma, G. (2009). "Normalized (Pointwise) Mutual Information in Collocation Extraction". In: *Proceedings of GSCL*, pp. 31–40.
- Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). "The TIGER Treebank". In: *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Vol. 168.
- Bullinaria, J. A. and Levy, J. P. (2007). "Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study". In: *Behavior Research Methods* 39.3, pp. 510–526.
- Bunescu, R. and Pasca, M. (2006). "Using Encyclopedic Knowledge for Named Entity Disambiguation". In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.
- Caruana, R. (1997). "Multitask Learning". In: *Machine Learning* 28.1, pp. 41–75.
- Castro, D. and New, J. (2016). *The Promise of Artificial Intelligence*. Tech. rep. Center for Data Innovation, pp. 1–48.
- Chakrabarti, K., Chaudhuri, S., Cheng, T., and Xin, D. (2012). "A Framework for Robust Discovery of Entity Synonyms". In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1384–1392.
- Chen, H., Branavan, S. R. K., Barzilay, R., and Karger, D. R. (2009). "Global Models of Document Structure Using Latent Permutations". In: *Proceedings of Human Language Technologies*:

- The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, pp. 371–379.
- Chen, Z., Tamang, S., Lee, A., Li, X., Lin, W.-P., Snover, M. G., Ariles, J., Passantino, M., and Ji, H. (2010). "CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description." In: *Text Analysis Conference*.
- Cheng, G. Y. (2004). "A Study of Clinical Questions Posed by Hospital Clinicians". In: *Journal of the Medical Library Association* 92.4, pp. 445–458.
- Cheng, T., Lauw, H. W., and Paparizos, S. (2011). "Entity Synonyms for Structured Web Search". In: *IEEE Transactions on Knowledge and Data Engineering* 24.10, pp. 1862–1875.
- Chieu, H. L. and Ng, H. T. (2002). "Named Entity Recognition: A Maximum Entropy Approach Using Global Information". In: *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1*. ACL, pp. 1–7.
- Chinchor, N. and Robinson, P. (1997). "MUC-7 Named Entity Task Definition". In: *Proceedings of the 7th Conference on Message Understanding*. Vol. 29, pp. 1–21.
- Chiu, J. P. and Nichols, E. (2016). "Named Entity Recognition with Bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4, pp. 357–370.
- Choi, F. Y. Y. (2000). "Advances in Domain Independent Linear Text Segmentation". In: *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*. ACL, pp. 26–33.
- Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge.
- Chuklin, A., Serdyukov, P., and De Rijke, M. (2013). "Using Intent Information to Model User Behavior in Diversified Search". In: *European Conference on Information Retrieval*. Springer, pp. 1–13.
- Cimiano, P., Schultz, A., Sizov, S., Sorg, P., and Staab, S. (2009). "Explicit versus Latent Concept Models for Cross-Language Information Retrieval". In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Vol. 9, pp. 1513–1518.
- Cohan, A., Deroncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). "A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 615–621.
- Cohen, D., Yang, L., and Croft, W. B. (2018). "WikiPassageQA: A Benchmark Collection for Research on Non-Factoid Answer Passage Retrieval". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, pp. 1165–1168.
- Collins, M. and Singer, Y. (1999). "Unsupervised Models for Named Entity Classification". In: *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). "Natural Language Processing (Almost) from Scratch". In: *Journal of Machine Learning Research* 12.Aug, pp. 2493–2537.



- Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. (2017). "Very Deep Convolutional Networks for Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Vol. 1. ACL, pp. 1107–1116.
- Cornolti, M., Ferragina, P., and Ciaramita, M. (2013). "A Framework for Benchmarking Entity-Annotation Systems". In: *Proceedings of the 22nd International Conference on World Wide Web*. ACM, pp. 249–260.
- Cucerzan, S. (2007). "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 708–716.
- Cunningham, H., Maynard, D., Bontcheva, K., Maynard, H., and Tablan, V (2002). "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, pp. 168–175.
- Curran, J. R. and Clark, S. (2003). "Language Independent NER Using a Maximum Entropy Tagger". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 164–167.
- Dai, Z., Xiong, C., Callan, J., and Liu, Z. (2018). "Convolutional Neural Networks for Soft-Matching n-Grams in Ad-Hoc Search". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, pp. 126–134.
- Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2020). "Plug and Play Language Models: A Simple Approach to Controlled Text Generation". In: *International Conference on Learning Representations*.
- de Masson d'Autume, C., Ruder, S., Kong, L., and Yogatama, D. (2019). "Episodic Memory in Lifelong Language Learning". In: *Advances in Neural Information Processing Systems*, pp. 13122–13131.
- de Saussure, F. (1916). *Cours de Linguistique Générale*. Paris: Payot.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). "Indexing by Latent Semantic Analysis". In: *Journal of the American Society for Information Science* 41.6, pp. 391–407.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1, pp. 4171–4186.
- Dias, G., Alves, E., and Lopes, J. G. P. (2007). "Topic Segmentation Algorithms for Text Summarization and Passage Retrieval: An Exhaustive Evaluation". In: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*. Vol. 7, pp. 1334–1340.
- Dieng, A. B., Wang, C., Gao, J., and Paisley, J. (2017). "TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency". In: *5th International Conference on Learning Representations*.

- Dojchinovski, M. and Kliegr, T. (2013). "Entityclassifier.eu: Real-Time Classification of Entities in Text with Wikipedia". In: *Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 654–658.
- Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. (2010). "Entity Disambiguation for Knowledge Base Population". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. ACL, pp. 277–285.
- Du, L., Buntine, W., and Johnson, M. (2013). "Topic Segmentation with a Structured Topic Model". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 190–200.
- Durrett, G. and Klein, D. (2014). "A Joint Model for Entity Analysis: Coreference, Typing, and Linking". In: *Transactions of the Association for Computational Linguistics 2*, pp. 477–490.
- Eisenstein, J. and Barzilay, R. (2008). "Bayesian Unsupervised Topic Segmentation". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. ACL, pp. 334–343.
- Etzioni, O., Banko, M., and Cafarella, M. J. (2006). "Machine Reading". In: *AAAI*. Vol. 6, pp. 1517–1519.
- Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). "Unsupervised Named-Entity Extraction from the Web: An Experimental Study". In: *Artificial intelligence* 165.1, pp. 91–134.
- Faro, S. and Lecroq, T. (2009). "Efficient Variants of the Backward-Oracle-Matching Algorithm". In: *International Journal of Foundations of Computer Science* 20.06, pp. 967–984.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.
- Ferragina, P. and Scaiella, U. (2010). "TAGME: On-the-Fly Annotation of Short Text Fragments (by Wikipedia Entities)". In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, pp. 1625–1628.
- Ferrucci, D. and Lally, A. (2004). "UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment". In: *Natural Language Engineering* 10.3-4, pp. 327–348.
- Firth, J. R. (1957). "A Synopsis of Linguistic Theory, 1930-1955". In: *Studies in Linguistic Analysis*.
- Forney, G. D. (1973). "The Viterbi Algorithm". In: *Proceedings of the IEEE* 61.3, pp. 268–278.
- Francis-Landau, M., Durrett, G., and Klein, D. (2016). "Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1256–1261.
- Fujiwara, T., Yamamoto, Y., Kim, J.-D., Buske, O., and Takagi, T. (2018). "PubCaseFinder: A Case-Report-Based, Phenotype-Driven Differential-Diagnosis System for Rare Diseases". In: *The American Journal of Human Genetics* 103.3, pp. 389–399.
- Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). "The Vocabulary Problem in Human-System Communication". In: *Communications of the ACM* 30.11, pp. 964–971.

- Gabrilovich, E. and Markovitch, S. (2007). "Computing Semantic Relatedness Using Wikipedia-Based Explicit Semantic Analysis". In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pp. 1606–1611.
- Gal, Y. and Ghahramani, Z. (2016). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *International Conference on Machine Learning*, pp. 1050–1059.
- Gantz, J. and Reinsel, D. (2012). *The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East*. Tech. rep. 1414. IDC, pp. 1–16.
- Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). "Convolutional Sequence to Sequence Learning". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70, pp. 1243–1252.
- Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). "Learning to Forget: Continual Prediction with LSTM". In: *Neural Computation* 12.10, pp. 2451–2471.
- Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldrige, J., Ie, E., and Garcia-Olano, D. (2019). "Learning Dense Representations for Entity Retrieval". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. ACL, pp. 528–537.
- Gillick, D., Presta, A., and Tomar, G. S. (2018). "End-to-End Retrieval in Continuous Space". In: *arXiv:1811.08008 [cs.IR]*.
- Glavaš, G., Nanni, F., and Ponzetto, S. P. (2016). "Unsupervised Text Segmentation Using Semantic Relatedness Graphs". In: *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. ACL, pp. 125–130.
- Goldberg, Y. (2016). "A Primer on Neural Network Models for Natural Language Processing". In: *Journal of Artificial Intelligence Research* 57, pp. 345–420.
- Gorman, P. N., Ash, J., and Wykoff, L. (1994). "Can Primary Care Physicians' Questions Be Answered Using the Medical Journal Literature?" In: *Bulletin of the Medical Library Association* 82.2.
- Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385. Berlin Heidelberg: Springer.
- Grishman, R. and Sundheim, B. M. (1996). "Message Understanding Conference-6: A Brief History". In: *The 16th International Conference on Computational Linguistics*.
- Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). "A Deep Relevance Matching Model for Ad-Hoc Retrieval". In: *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, pp. 55–64.
- Guo, J., Fan, Y., Ji, X., and Cheng, X. (2019). "MatchZoo: A Learning, Practicing, and Developing System for Neural Text Matching". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 1297–1300.
- Gupta, N., Singh, S., and Roth, D. (2017). "Entity Linking via Joint Encoding of Types, Descriptions, and Context". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2681–2690.

- Gutmann, M. U. and Hyvärinen, A. (2012). "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics". In: *Journal of Machine Learning Research* 13.Feb, pp. 307–361.
- Hachey, B., Radford, W., Nothman, J., Honnibal, M., and Curran, J. R. (2013). "Evaluating Entity Linking with Wikipedia". In: *Artificial Intelligence* 194, pp. 130–150.
- Han, X. and Zhao, J. (2009). "NLPR\_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking." In: *Text Analysis Conference*.
- Hanauer, D. A., Mei, Q., Law, J., Khanna, R., and Zheng, K. (2015). "Supporting Information Retrieval from Electronic Health Records: A Report of University of Michigan's Nine-Year Experience in Developing and Using the Electronic Medical Record Search Engine (EMERSE)". In: *Journal of Biomedical Informatics* 55, pp. 290–300.
- Harris, Z. S. (1954). "Distributional Structure". In: *WORD* 10.2-3, pp. 146–162.
- He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., and Wang, H. (2013). "Learning Entity Representation for Entity Disambiguation". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Vol. 2 (Short Papers), pp. 30–34.
- Hearst, M. A. (1997). "TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages". In: *Computational Linguistics* 23.1, pp. 33–64.
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). "Support Vector Machines". In: *IEEE Intelligent Systems and their Applications* 13.4, pp. 18–28.
- Heckmann, I., Comes, T., and Nickel, S. (2015). "A Critical Review on Supply Chain Risk-Definition, Measure and Modeling". In: *Omega* 52, pp. 119–132.
- Herbrich, R. (2000). "Large Margin Rank Boundaries for Ordinal Regression". In: *Advances in Large Margin Classifiers*, pp. 115–132.
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). "Teaching Machines to Read and Comprehend". In: *Advances in Neural Information Processing Systems*, pp. 1693–1701.
- Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M., and Berthelot, D. (2016). "WikiReading: A Novel Large-Scale Language Understanding Task over Wikipedia". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Vol. 1. ACL, pp. 1535–1545.
- Hill, F., Cho, K., and Korhonen, A. (2016). "Learning Distributed Representations of Sentences from Unlabelled Data". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1367–1377.
- Hirschberg, J. and Manning, C. D. (2015). "Advances in Natural Language Processing". In: *Science* 349.6245, pp. 261–266.
- Hoa T. Le, Cerisara, C., and Denis, A. (2018). "Do Convolutional Networks Need to Be Deep for Text Classification?" In: *Association for the Advancement of Artificial Intelligence 2018 Workshop on Affective Content Analysis*, pp. 29–36.
- Hochreiter, S. and Schmidhuber, J. (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.

- Hoffart, J., Seufert, S., Nguyen, D. B., Theobald, M., and Weikum, G. (2012). "KORE: Keyphrase Overlap Relatedness for Entity Disambiguation". In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, pp. 545–554.
- Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). "YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. AAAI Press, pp. 3161–3165.
- Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). "Robust Disambiguation of Named Entities in Text". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, pp. 782–792.
- Holley, J. O., Luehr, P. H., Smith, J. R., and Schwerha IV, J. J. (2010). "Electronic Discovery". In: *Handbook of Digital Forensics and Investigation*. Elsevier, pp. 63–133.
- Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). "OntoNotes: The 90% Solution". In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 57–60.
- Hu, B., Lu, Z., Li, H., and Chen, Q. (2014). "Convolutional Neural Network Architectures for Matching Natural Language Sentences". In: *Advances in Neural Information Processing Systems*, pp. 2042–2050.
- Hu, Z., Huang, P., Deng, Y., Gao, Y., and Xing, E. (2015). "Entity Hierarchy Embedding". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Vol. 1, pp. 1292–1300.
- Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. (2013). "Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data". In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. ACM, pp. 2333–2338.
- Huang, X., Peng, F., Schuurmans, D., Cercone, N., and Robertson, S. E. (2003). "Applying Machine Learning to Text Segmentation for Information Retrieval". In: *Information Retrieval* 6.3-4, pp. 333–362.
- Huang, X., Lin, J., and Demner-Fushman, D. (2006). "Evaluation of PICO as a Knowledge Representation for Clinical Questions". In: *AMIA Annual Symposium Proceedings*. Vol. 2006. AMIA, pp. 359–363.
- Huang, Z., Xu, W., and Yu, K. (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging". In: *arXiv:1508.01991 [cs.CL]*.
- Huber, P. J. (1992). "Robust Estimation of a Location Parameter". In: *Breakthroughs in Statistics*. Springer, pp. 492–518.
- Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. (1998). "University of Sheffield: Description of the LaSIE-II System as Used for MUC-7". In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.
- Inmon, W. H., Linstedt, D., and Levins, M. (2019). *Data Architecture: A Primer for the Data Scientist: A Primer for the Data Scientist*. Academic Press.

- INSERM (1997). *Orphanet: An Online Database of Rare Diseases and Orphan Drugs*. Tech. rep. Copyright, INSERM.
- Jeong, M. and Titov, I. (2010). "Multi-Document Topic Segmentation". In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, pp. 1119–1128.
- Ji, H. and Grishman, R. (2011). "Knowledge Base Population: Successful Approaches and Challenges". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. ACL, pp. 1148–1158.
- Jiang, B. (2012). "Head/Tail Breaks: A New Classification Scheme for Data with a Heavy-Tailed Distribution". In: *The Professional Geographer* 65.3, pp. 482–494.
- Jin, H., Schwartz, R., Sista, S., and Walls, F. (1999). "Topic Tracking for Radio, TV Broadcast and Newswire". In: *Proceedings of the DARPA Broadcast News Workshop*. Morgan Kaufmann, pp. 199–204.
- Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. (2019). "PubMedQA: A Dataset for Biomedical Research Question Answering". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2567–2577.
- Joachims, T. (1998). "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". In: *European Conference on Machine Learning*. Springer, pp. 137–142.
- Jones, S. K. (1972). "A Statistical Interpretation of Term Specificity and Its Application to Retrieval". In: *Journal of Documentation* 28.1, pp. 11–21.
- Jurafsky, D. and Martin, J. H. (2019). *Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Vol. Third Edition draft.
- Karttunen, L. (1976). "Discourse Referents". In: *Notes from the Linguistic Underground*. Brill, pp. 363–385.
- Keikha, M., Park, J. H., Croft, W. B., and Sanderson, M. (2014). "Retrieving Passages and Finding Answers". In: *Proceedings of the 2014 Australasian Document Computing Symposium*. ACM, pp. 81–84.
- Kendall, A. and Gal, Y. (2017). "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*, pp. 5574–5584.
- Kenter, T. and de Rijke, M. (2015). "Short Text Similarity with Word Embeddings". In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. ACM, pp. 1411–1420.
- Kim, J.-D., Ohta, T., Tateisi, Y., and Tsujii, J. (Mar. 2003). "GENIA Corpus – a Semantically Annotated Corpus for Bio-Textmining". In: *Bioinformatics* 19.suppl 1, pp. i180–i182.
- Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004). "Introduction to the Bio-Entity Recognition Task at JNLPBA". In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, pp. 70–75.

- Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751.
- Kingma, D. and Ba, J. (2015). "ADAM: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations*.
- Kingsbury, P. and Palmer, M. (2002). "From TreeBank to PropBank". In: *Language Resources and Evaluation*, pp. 1989–1993.
- Koshorek, O., Cohen, A., Mor, N., Rotman, M., and Berant, J. (2018). "Text Segmentation as a Supervised Learning Task". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 2, pp. 469–473.
- Krishnan, V. and Manning, C. D. (2006). "An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. ACL, pp. 1121–1128.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.
- Krupka, G. and Hausman, K. (1998). "IsoQuest Inc.: Description of the NetOwl™ Extractor System as Used for MUC-7". In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia*.
- Kuhlthau, C. C. (1991). "Inside the Search Process: Information Seeking from the User's Perspective". In: *Journal of the American Society for Information Science* 42.5, pp. 361–371.
- Kumaran, G. and Allan, J. (2004). "Text Classification and Named Entities for New Event Detection". In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 297–304.
- Kümmel, C. (2018). "Learning a Sampling Strategy for Named Entity Recognition". Master Thesis. Berlin: Beuth Hochschule für Technik.
- Lafferty, J., McCallum, A., and Pereira, F. C. (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the 18th International Conference on Machine Learning 2001*, pp. 282–289.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). "Neural Architectures for Named Entity Recognition". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270.
- Landauer, T. K. and Dumais, S. T. (1997). "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge." In: *Psychological review* 104.2.
- Le, Q. V. and Mikolov, T. (2014). "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32, pp. 1188–1196.

- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4, pp. 541–551.
- LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). "A Tutorial on Energy-Based Learning". In: *Predicting Structured Data*. Vol. 1. MIT Press.
- Lee, E. G. and Willging, T. E. (2010). "Defining the Problem of Cost in Federal Civil Litigation". In: *Duke Law Journal* 60.3, pp. 765–788.
- Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). "BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining". In: *Bioinformatics*, pp. 1–7.
- Leetaru, K. and Schrod, P. A. (2013). "GDELT: Global Data on Events, Location, and Tone, 1979–2012". In: *ISA Annual Convention*. Vol. 2, pp. 1–49.
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., and Bizer, C. (2015). "DBpedia – a Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia". In: *Semantic Web* 6.2, pp. 167–195.
- Li, J., Sun, A., Han, J., and Li, C. (2020). "A Survey on Deep Learning for Named Entity Recognition". In: *IEEE Transactions on Knowledge and Data Engineering*.
- Li, L., Chen, S., Kleban, J., and Gupta, A. (2015). "Counterfactual Estimation and Optimization of Click Metrics in Search Engines: A Case Study". In: *Proceedings of the 24th International Conference on World Wide Web*, pp. 929–934.
- Ling, X., Singh, S., and Weld, D. S. (2015). "Design Challenges for Entity Linking". In: *Transactions of the Association for Computational Linguistics* 3, pp. 315–328.
- Ling, X. and Weld, D. S. (2012). "Fine-Grained Entity Recognition". In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- Lipton, Z. C. and Berkowitz, J. (2015). "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *arXiv:1506.00019 [cs.LG]*.
- Liu, J., Ren, X., Shang, J., Cassidy, T., Voss, C. R., and Han, J. (2016). "Representing Documents via Latent Keyphrase Inference". In: *Proceedings of the 25th International Conference on World Wide Web*, pp. 1057–1067.
- Liu, X. and Croft, W. B. (2002). "Passage Retrieval Based on Language Models". In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*. ACM, pp. 375–382.
- Logeswaran, L., Chang, M.-W., Lee, K., Toutanova, K., Devlin, J., and Lee, H. (2019). "Zero-Shot Entity Linking by Reading Entity Descriptions". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3449–3460.
- Loper, E. and Bird, S. (2002). "NLTK: The Natural Language Toolkit". In: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp. 63–70.
- Löser, A., Arnold, S., and Fiehn, T. (2012). "The GoOLAP Fact Retrieval Framework". In: *Business Intelligence*. Springer, pp. 84–97.



- Loshchilov, I. and Hutter, F. (2017). “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*.
- Lund, K. and Burgess, C. (1996). “Producing High-Dimensional Semantic Spaces from Lexical Co-Occurrence”. In: *Behavior Research Methods, Instruments, & Computers* 28.2, pp. 203–208.
- Ma, X. and Hovy, E. (2016). “End-to-End Sequence Labeling via Bi-Directional LSTM-CNNs-CRF”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Vol. 1. ACL, pp. 1064–1074.
- MacAvaney, S., Yates, A., Cohan, A., Soldaini, L., Hui, K., Goharian, N., and Frieder, O. (2018). “Characterizing Question Facets for Complex Answer Retrieval”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM Press, pp. 1205–1208.
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). “The Stanford CoreNLP Natural Language Processing Toolkit”. In: *ACL System Demonstrations*, pp. 55–60.
- Maqsud, U., Arnold, S., Hülfenhaus, M., and Akbik, A. (2014). “Nerdle: Topic-Specific Question Answering Using Wikia Seeds”. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pp. 81–85.
- Marchionini, G. (2006). “Exploratory Search: From Finding to Understanding”. In: *CACM* 49.4, pp. 41–46.
- Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). “Building a Large Annotated Corpus of English: The Penn Treebank”. In: *Computational Linguistics* 19.2.
- MarketsandMarkets (2019). *eDiscovery Market by Component (Software (Processing, Review and Analysis, Identification, Preservation and Collection, and Production and Presentation) and Services), Deployment Type, Organization Size, Vertical, and Region - Global Forecast to 2023*. Tech. rep. MarketsandMarkets.
- McCallum, A., Freitag, D., and Pereira, F. C. (2000). “Maximum Entropy Markov Models for Information Extraction and Segmentation.” In: *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 591–598.
- McCallum, A. and Li, W. (2003). “Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. Stroudsburg, PA, USA, pp. 188–191.
- McNamee, P. and Mayfield, J. (2002). “Entity Extraction without Language-Specific Resources”. In: *Proceedings of the 6th Conference on Natural Language Learning-Volume 20*. ACL, pp. 1–4.
- Mehlitz, R. (2019). “Deep-Learning für das Erkennen und Verlinken in Texten für die Automobilbranche”. German. Master thesis. Berlin: Beuth Hochschule für Technik.

- Mendes, P. N., Jakob, M., Garcia-Silva, A., and Bizer, C. (2011). "DBpedia Spotlight: Shedding Light on the Web of Documents". In: *Proceedings of the 7th International Conference on Semantic Systems*. ACM, pp. 1–8.
- Mihalcea, R. and Csomai, A. (2007). "Wikify! Linking Documents to Encyclopedic Knowledge". In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pp. 233–242.
- Mikheev, A., Moens, M., and Grover, C. (1999). "Named Entity Recognition without Gazetteers". In: *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*. ACL, pp. 1–8.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). "Efficient Estimation of Word Representations in Vector Space". In: *International Conference on Learning Representations*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). "Distributed Representations of Words and Phrases and Their Compositionality". In: *Advances in Neural Information Processing Systems*, pp. 3111–3119.
- Milne, D. and Witten, I. H. (2008). "Learning to Link with Wikipedia". In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 509–518.
- Mitchell, A., Strassel, S., Huang, S., and Zakhary, R. (2005). *ACE 2004 Multilingual Training Corpus*. <https://catalog.ldc.upenn.edu/LDC2005T09>. (accessed 2016-05-23).
- Mitchell, J. and Lapata, M. (2010). "Composition in Distributional Models of Semantics". In: *Cognitive Science* 34.8, pp. 1388–1429.
- Mitra, B. and Craswell, N. (2018). "An Introduction to Neural Information Retrieval". In: *Foundations and Trends in Information Retrieval* 13.1, pp. 1–126.
- Mitra, B., Diaz, F., and Craswell, N. (2017). "Learning to Match Using Local and Distributed Representations of Text for Web Search". In: *Proceedings of the 26th International Conference on World Wide Web*. IW3C2, pp. 1291–1299.
- Moro, A., Raganato, A., and Navigli, R. (2014). "Entity Linking Meets Word Sense Disambiguation: A Unified Approach". In: *Transactions of the Association for Computational Linguistics* 2, pp. 231–244.
- Nadeau, D., Turney, P. D., and Matwin, S. (2006). "Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity". In: *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, pp. 266–277.
- Naili, M., Chaïbi, A. H., and Ghézala, H. H. B. (2017). "Comparative Study of Word Embedding Methods in Topic Segmentation". In: *Proceedings of the 21st International Conference Knowledge-Based and Intelligent Information & Engineering Systems*. Vol. 112. Procedia Computer Science. Elsevier, pp. 340–349.
- Nair, V. and Hinton, G. E. (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 807–814.

- Nanni, F., Mitra, B., Magnusson, M., and Dietz, L. (2017). "Benchmark for Complex Answer Retrieval". In: *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, pp. 293–296.
- Nanni, F., Ponzetto, S. P., and Dietz, L. (2018). "Entity-Aspect Linking: Providing Fine-Grained Semantics of Entities in Context". In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. ACM, pp. 49–58.
- Navigli, R. and Ponzetto, S. P. (2012). "BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network". In: *Artificial Intelligence* 193, pp. 217–250.
- Newman, M. E. J. (2006). "Finding Community Structure in Networks Using the Eigenvectors of Matrices". In: *Physical Review E* 74.3, p. 036104.
- Nickel, M. and Kiela, D. (2017). "Poincaré Embeddings for Learning Hierarchical Representations". In: *Advances in Neural Information Processing Systems*, pp. 6338–6347.
- Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). "Universal Dependencies v1: A Multilingual Treebank Collection". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1659–1666.
- O'Connor, J. (1980). "Answer-Passage Retrieval by Text Searching". In: *Journal of the American Society for Information Science* 31.4, pp. 227–239.
- Onal, K. D., Zhang, Y., Altingovde, I. S., Rahman, M. M., Karagoz, P., Braylan, A., Dang, B., Chang, H.-L., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., McDonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., de Rijke, M., and Lease, M. (2018). "Neural Information Retrieval: At the End of the Early Years". In: *Information Retrieval Journal* 21.2-3, pp. 111–182.
- Paisley, J., Blei, D. M., and Jordan, M. I. (2012). "Variational Bayesian Inference with Stochastic Search". In: *Proceedings of the 29th International Conference on Machine Learning*, pp. 1363–1370.
- Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. (2016). "Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval". In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24.4, pp. 694–707.
- Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). "Zero-Shot Learning with Semantic Output Codes". In: *Advances in Neural Information Processing Systems*, pp. 1410–1418.
- Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., and Cheng, X. (2016). "Text Matching as Image Recognition". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2793–2799.
- Papaioannou, J.-M., Arnold, S., Gers, F. A., Löser, A., Mayrdorfer, M., and Budde, K. (2020). "Aspect-Based Passage Retrieval with Contextualized Discourse Vectors". In: *[IN SUBMISSION] COLING 2020 System Demonstrations*.

- Pappu, A., Blanco, R., Mehdad, Y., Stent, A., and Thadani, K. (2017). "Lightweight Multilingual Entity Extraction and Linking". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM Press, pp. 365–374.
- Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). "Continual Lifelong Learning with Neural Networks: A Review". In: *Neural Networks* 113, pp. 54–71.
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). "On the Difficulty of Training Recurrent Neural Networks". In: *International Conference on Machine Learning*, pp. 1310–1318.
- Peng, F., Feng, F., and McCallum, A. (2004). "Chinese Segmentation and New Word Detection Using Conditional Random Fields". In: *Proceedings of the 20th International Conference on Computational Linguistics*. ACL.
- Pennington, J., Socher, R., and Manning, C. D. (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing*, pp. 1532–1543.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1, pp. 2227–2237.
- Piccardi, T., Catasta, M., Zia, L., and West, R. (2018). "Structuring Wikipedia Articles with Section Recommendations". In: *Proceedings of the 41th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 665–674.
- Pink, G., Nothman, J., and Curran, J. R. (2014). "Analysing Recall Loss in Named Entity Slot Filling". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 820–830.
- Prabhu, Y. and Varma, M. (2014). "FastXML: A Fast, Accurate and Stable Tree-Classifer for Extreme Multi-Label Learning". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 263–272.
- Prokofyev, R., Demartini, G., and Cudré-Mauroux, P. (2014). "Effective Named Entity Recognition for Idiosyncratic Web Collections". In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 397–408.
- Quinlan, J. R. (1986). "Induction of Decision Trees". In: *Machine Learning* 1.1, pp. 81–106.
- Rabiner, L. R. (1989). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). "Improving Language Understanding by Generative Pre-Training". In:
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392.
- Ranganath, R., Gerrish, S., and Blei, D. M. (2014). "Black Box Variational Inference". In: *Journal of Machine Learning Research* 33, pp. 814–822.

- Ratinov, L. and Roth, D. (2009). "Design Challenges and Misconceptions in Named Entity Recognition". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. ACL, pp. 147–155.
- Ratinov, L., Roth, D., Downey, D., and Anderson, M. (2011). "Local and Global Algorithms for Disambiguation to Wikipedia". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. ACL, pp. 1375–1384.
- Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2020). "Snorkel: Rapid Training Data Creation with Weak Supervision". In: *The VLDB Journal* 29.2, pp. 709–730.
- Reinsel, D, Gantz, J, and Rydning, J (2018). *Data Age 2025: The Digitization of the World from Edge to Core*. IDC White Paper US44413318. IDC.
- Reports and Data (2020). *Clinical Decision Support System (CDSS) Market By Component, By Product, By Type, By Mode of Delivery, By Level of Interactivity, By Setting Outlook, By Usage, By Application, By End-Use, And Segment Forecasts, 2017-2027*. Tech. rep. Reports and Data.
- Richardson, W. S., Wilson, M. C., Nishikawa, J., and Hayward, R. S. (1995). "The Well-Built Clinical Question: A Key to Evidence-Based Decisions". In: *ACP Journal Club* 123.3, A12–3.
- Riedl, M. and Biemann, C. (2012). "TopicTiling: A Text Segmentation Algorithm Based on LDA". In: *Proceedings of ACL 2012 Student Research Workshop*. ACL, pp. 37–42.
- Robertson, S. E. and Jones, K. S. (1976). "Relevance Weighting of Search Terms". In: *Journal of the American Society for Information Science* 27.3, pp. 129–146.
- Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). "Okapi at TREC-3". In: *NIST Special Publication SP 109*.
- Sahlgren, M. (2008). "The Distributional Hypothesis". In: *Italian Journal of Linguistics* 20.1, pp. 33–54.
- Salton, G., Allan, J., and Buckley, C. (1993). "Approaches to Passage Retrieval in Full Text Information Systems". In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 49–58.
- Salton, G. and Buckley, C. (1988). "Term-Weighting Approaches in Automatic Text Retrieval". In: *Information Processing & Management* 24.5, pp. 513–523.
- Salton, G., Wong, A., and Yang, C.-S. (1975). "A Vector Space Model for Automatic Indexing". In: *Communications of the ACM* 18.11, pp. 613–620.
- Santos, C. N. dos, Xiang, B., and Zhou, B. (2015). "Classifying Relations by Ranking with Convolutional Neural Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. ACL, pp. 626–634.
- Sarawagi, S. (2008). "Information Extraction". In: *Foundations and Trends in Databases* 1.3, pp. 261–377.
- Saxe, A. M., McClellans, J. L., and Ganguli, S. (2013). "Learning Hierarchical Categories in Deep Neural Networks". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 35.

- Schardt, C., Adams, M. B., Owens, T., Keitz, S., and Fontelo, P. (2007). "Utilization of the PICO Framework to Improve Searching PubMed for Clinical Questions". In: *BMC Medical Informatics and Decision Making* 7.1.
- Schneider, R., Arnold, S., Oberhauser, T., Klatt, T., Steffek, T., and Löser, A. (2018). "Smart-MD: Neural Paragraph Retrieval of Medical Topics". In: *The Web Conference 2018 Companion*. IW3C2, pp. 203–206.
- Schütze, H. (1993). "Word Space". In: *Advances in Neural Information Processing Systems*. Vol. 5.
- Schütze, H. and Pedersen, J. (1993). "A Vector Model for Syntagmatic and Paradigmatic Relatedness". In: *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*. Oxford, pp. 104–113.
- Sehikh, I., Fohr, D., and Illina, I. (2017). "Topic Segmentation in ASR Transcripts Using Bidirectional RNNs for Change Detection". In: *Automatic Speech Recognition and Understanding Workshop*. IEEE, pp. 512–518.
- Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2017). "Bidirectional Attention Flow for Machine Comprehension". In: *5th International Conference on Learning Representations*.
- Serrà, J. and Karatzoglou, A. (2017). "Getting Deep Recommenders Fit: Bloom Embeddings for Sparse Binary Input/Output Networks". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, pp. 279–287.
- Settles, B. (2010). *Active Learning Literature Survey*. Tech. rep.
- Shen, W., Wang, J., and Han, J. (2015). "Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions". In: *IEEE Transactions on Knowledge and Data Engineering* 27.2, pp. 443–460.
- Shen, W., Wang, J., Luo, P., and Wang, M. (2012). "LINDEN: Linking Named Entities with Knowledge Base via Semantic Knowledge". In: *Proceedings of the 21st International Conference on World Wide Web*, pp. 449–458.
- Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). "Learning Semantic Representations Using Convolutional Neural Networks for Web Search". In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM, pp. 373–374.
- Sil, A., Kundu, G., Florian, R., and Hamza, W. (2018). "Neural Cross-Lingual Entity Linking". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Speck, R. and Ngomo, A.-C. N. (2014). "Ensemble Learning for Named Entity Recognition". In: *International Semantic Web Conference*. Springer, pp. 519–534.
- Sullivan, C. C. (2017). *What a Million-Dollar eDiscovery Bill Looks Like*. <https://www.logikcull.com/blog/million-dollar-ediscovery-bill-looks-like>. (accessed 2020-04-06).
- Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., and Wang, X. (2015). "Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

- Szarvas, G., Farkas, R., and Kocsor, A. (2006). "A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms". In: *International Conference on Discovery Science*. Springer, pp. 267–278.
- Taneva, B., Cheng, T., Chakrabarti, K., and He, Y. (2013). "Mining Acronym Expansions and Their Meanings Using Query Click Log". In: *Proceedings of the 22nd International Conference on World Wide Web*, pp. 1261–1272.
- Tanon, T. P., Vrandečić, D., Schaffert, S., Steiner, T., and Pintscher, L. (2016). "From Freebase to Wikidata: The Great Migration". In: *Proceedings of the 25th International Conference on World Wide Web*, pp. 1419–1428.
- Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton, G. (2003). "Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM, pp. 41–47.
- Tepper, M., Capurro, D., Xia, F., Vanderwende, L., and Yetisgen-Yildiz, M. (2012). "Statistical Section Segmentation in Free-Text Clinical Records". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pp. 2001–2008.
- Tieleman, T. and Hinton, G. (2012). "Lecture 6.5 RMSProp: Divide the Gradient by a Running Average of Its Recent Magnitude". In: *Coursera: Neural Networks for Machine Learning 4.2*.
- Toms, E. G., O'Brien, H. L., Kopak, R., and Freund, L. (2005). "Searching for Relevance in the Relevance of Search". In: *International Conference on Conceptions of Library and Information Sciences*. Springer, pp. 59–78.
- Tsatsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Androutsopoulos, I., Gaussier, E., Gallinari, P., Artieres, T., Alvers, M. R., and Zschunke, M. (2012). "BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering". In: *AAAI Technical Report FS-12-05 Information Retrieval and Knowledge Discovery in Biomedical Text*, pp. 92–98.
- Tsoumakas, G., Katakis, I., and Vlahavas, I. (2009). "Mining Multi-Label Data". In: *Data Mining and Knowledge Discovery Handbook*. Springer, pp. 667–685.
- Turney, P. D. and Pantel, P. (2010). "From Frequency to Meaning: Vector Space Models of Semantics". In: *Journal of Artificial Intelligence Research* 37, pp. 141–188.
- Umbenhauer, B., Flynn, R. P., and Mitchell, P. (2019). *Complexity: Overcoming Obstacles and Seizing Opportunities. The Deloitte Global Chief Procurement Officer Survey 2019*. Tech. rep. Deloitte LLC.
- Umbenhauer, B. and Younger, L. (2018). *Leadership: Driving Innovation and Delivering Impact. The Deloitte Global Chief Procurement Officer Survey 2018*. Tech. rep. Deloitte LLP and Odgers Berndtson.
- Usbeck, R., Röder, M., Ngonga Ngomo, A.-C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., Ferragina, P., Lemke, C., Moro, A., Navigli, R., Piccinno, F., Rizzo, G., Sack, H., Speck, R., Troncy, R., Waitelonis, J., and Wesemann, L. (2015). "GERBIL: General Entity Annotator Benchmarking Framework". In: *Proceedings of the 24th International Conference on World Wide Web*. IW3C2, pp. 1133–1143.

- Utiyama, M. and Isahara, H. (2001). "A Statistical Model for Domain-Independent Text Segmentation". In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. ACL, pp. 499–506.
- Uzuner, Ö., South, B. R., Shen, S., and DuVall, S. L. (2011). "2010 I2b2/VA Challenge on Concepts, Assertions, and Relations in Clinical Text". In: *Journal of the American Medical Informatics Association* 18.5, pp. 552–556.
- Van Erp, M., Rizzo, G., and Troncy, R. (2013). "Learning with the Web: Spotting Named Entities on the Intersection of NERD and Machine Learning." In: *Making Sense of Microposts*. ACM, pp. 27–30.
- Varma, V., Bharat, V., Kovelamudi, S., Bysani, P., GSK, S., N, K. K., Reddy, K., Kumar, K., and Maganti, N. (2009). "IIIT Hyderabad at TAC 2009". In: *Text Analysis Conference*.
- Vassiliadis, P. (2009). "A Survey of Extract–Transform–Load Technology". In: *International Journal of Data Warehousing and Mining (IJDWM)* 5.3, pp. 1–27.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.
- Vrandečić, D. and Krötzsch, M. (2014). "Wikidata: A Free Collaborative Knowledgebase". In: *Communications of the ACM* 57.10, pp. 78–85.
- Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., and Cheng, X. (2016). "A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2835–2841.
- Wang, L., Li, S., Lyu, Y., and Wang, H. (2017a). "Learning to Rank Semantic Coherence for Topic Segmentation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL, pp. 1340–1344.
- Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R. M., Liu, Z., Merrill, W., Mooney, P., Murdick, D. A., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A. D., Wang, K., Wilhelm, C., Xie, B., Raymond, D. M., Weld, D. S., Etzioni, O., and Kohlmeier, S. (2020). "CORD-19: The Covid-19 Open Research Dataset". In: *arXiv:2004.10706 [cs.DL]*.
- Wang, W., Yang, N., Wei, F., Chang, B., and Zhou, M. (2017b). "Gated Self-Matching Networks for Reading Comprehension and Question Answering". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 189–198.
- Werbos, P. J. (1990). "Backpropagation Through Time: What It Does And How To Do It". In: *Proceedings of the IEEE* 78.10, pp. 1550–1560.
- White, R. W. and Horvitz, E. (2014). "From Health Search to Healthcare: Explorations of Intention and Utilization via Query Logs and User Surveys." In: *Journal of the American Medical Informatics Association* 21.1, pp. 49–55.
- Williams, R. J. and Zipser, D. (1989). "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: *Neural Computation* 1.2, pp. 270–280.



- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *arXiv:1609.08144 [cs.CL]*.
- Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. (2017). "End-to-End Neural Ad-Hoc Ranking with Kernel Pooling". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 55–64.
- Yang, L., Ai, Q., Guo, J., and Croft, W. B. (2016a). "aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 287–296.
- Yang, L., Ai, Q., Spina, D., Chen, R.-C., Pang, L., Croft, W. B., Guo, J., and Scholer, F. (2016b). "Beyond Factoid QA: Effective Methods for Non-Factoid Answer Sentence Retrieval". In: *European Conference on Information Retrieval*. Springer, pp. 115–128.
- Yeh, J.-F., Tan, Y.-S., and Lee, C.-H. (2016). "Topic Detection and Tracking for Conversational Content by Using Conceptual Dynamic Latent Dirichlet Allocation". In: *Neurocomputing* 216, pp. 310–318.
- Yogatama, D., d'Áutume, C. d. M., Connor, J., Kocisky, T., Chrzanowski, M., Kong, L., Lazaridou, A., Ling, W., Yu, L., Dyer, C., and Blunsom, P. (2019). "Learning and Evaluating General Linguistic Intelligence". In: *arXiv:1901.11373 [cs.LG]*.
- Yoo, I. and Mosa, A. S. M. (2015). "Analysis of PubMed User Sessions Using a Full-Day PubMed Query Log: A Comparison of Experienced and Nonexperienced PubMed Users". In: *JMIR Medical Informatics* 3.3.
- Zhang, M.-L. and Zhou, Z.-H. (2006). "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization". In: *IEEE transactions on Knowledge and Data Engineering* 18.10, pp. 1338–1351.
- Zhang, W., Sim, Y.-C., Su, J., and Tan, C.-L. (2011). "Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling". In: *Twenty-Second International Joint Conference on Artificial Intelligence*.
- Zheng, Z., Li, F., Huang, M., and Zhu, X. (2010). "Learning to Link Entities with Knowledge Base". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, pp. 483–491.
- Zhu, M., Ahuja, A., Wei, W., and Reddy, C. K. (2019). "A Hierarchical Attention Retrieval Model for Healthcare Question Answering". In: *The World Wide Web Conference*. ACM, pp. 2472–2482.
- Ziou, D. and Tabbone, S. (1998). "Edge Detection Techniques – An Overview". In: *Pattern Recognition and Image Analysis* 8, pp. 537–559.



# Sebastian Arnold

Horst-Kohl-Str. 16, 12157 Berlin, Germany  
phone +49 160 80 21 43 6  
mail@sebastian-arnold.net

Date and place of birth: July 10, 1984, Freiburg im Breisgau  
Citizenship: German

## Education and Qualifications

- 08/2016 – present    Doctoral Program in Computer Science**  
*Université de Fribourg, Switzerland*  
**Thesis:** “Neural Machine Reading for Domain-Specific Text Resources”, supervised by Prof. Dr. Philippe Cudré-Mauroux and Prof. Dr.-ing. habil. Alexander Löser.
- 10/2011 – 03/2015    Master of Science in Computer Science (M.Sc.-Inf.)**  
*Technische Universität Berlin, Germany*  
**Thesis:** “Interactive Classification of Keyword Search Queries”, supervised by Prof. Dr. rer. nat. Volker Markl and Prof. Dr.-ing. habil. Alexander Löser.
- 10/2006 – 09/2011    Bachelor of Science in Computer Science (B.Sc.-Inf.)**  
*Technische Universität Berlin, Germany*  
**Thesis:** “GoOLAP User Interaktion”, supervised by Prof. Dr. rer. nat. Volker Markl and Dr.-ing. Alexander Löser.
- 10/2005 – 09/2006    Communication in Social and Economic Contexts**  
*Berlin University of Arts (UdK), Germany (no formal degree)*
- 09/2001 – 07/2004    General Qualification for University Entrance**  
*Technisches Gymnasium Freiburg, Germany*

## Work Experience

- 07/2020 – present    Machine Learning Expert**  
*Curalie GmbH, Berlin, Germany*
- 06/2015 – 06/2020    Research Assistant**  
*Beuth University of Applied Sciences Berlin, Germany*  
*Data Science and Text-based Information Systems group (DATEXIS)*
- 04/2011 – 07/2014    Student Research Assistant**  
*Technische Universität Berlin, Germany*  
*Database Systems and Information Management group (DIMA)*
- 08/2005 – 08/2007    Student Assistant**  
*Dornier Consulting Berlin, Germany*

## **Languages**

- **German** – native speaker,
- **English** – highly proficient in spoken and written English,
- **French** – basic communication skills.

## **Research Interests**

- **Neural Machine Reading** – focused on distributional representations for long documents, topic and discourse modeling, self-supervised model training;
- **Information Extraction** – focused on domain-specific named entity recognition (NER) and linking (NEL), clinical information extraction, sparse training data;
- **Information Retrieval** – focused on neural answer passage retrieval from long documents;
- **Machine Learning** – focused on deep neural networks, e.g. LSTM, CNN, Transformer.

## **Personal Interests**

- **Music and Audio Technology** – drums, synthesizers, composition, sound design, digital signal processing (DSP), audio production and recording;
- **Embedded Systems** – hardware and software prototyping, sensors, communication;
- **Open Source Contributions** – maintainer of TeXoo Java IE framework. Contributions to Deeplearning4j, Apache Mahout and JUCE framework. See: [github.com/sebastianarnold](https://github.com/sebastianarnold).

## **Selected Publications**

- [S. Arnold](#), B. van Aken, P. Grundmann, F. A. Gers and A. Löser. Learning Contextualized Document Representations for Healthcare Answer Retrieval. *Proceedings of The Web Conference 2020*. ACM, 2020: 1332–1343.
- [S. Arnold](#), R. Schneider, P. Cudré-Mauroux, F. A. Gers and A. Löser. SECTOR: A Neural Model for Coherent Topic Segmentation and Classification. *Transactions of the Association for Computational Linguistics (TACL)* Vol. 7. MIT Press, 2019: 169–184.
- R. Schneider, [S. Arnold](#), T. Oberhauser, T. Klatt, T. Steffek and A. Löser. Smart-MD: Neural Paragraph Retrieval of Medical Topics. *World Wide Web Conference (Companion)*. IW3C2, 2018: 203–206.
- [S. Arnold](#), R. Dziuba and A. Löser. TASTY: Interactive Entity Linking As-You-Type. *26th International Conference on Computational Linguistics (COLING'16): Demos*. ACL, 2016: 111–115.
- [S. Arnold](#), F. A. Gers, T. Kiliyas and A. Löser. Robust Named Entity Recognition in Idiosyncratic Domains. *arXiv:1608.06757 [cs.CL]* 2016.
- [S. Arnold](#), A. Löser, and T. Kiliyas. Resolving Common Analytical Tasks in Text Databases. *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP (DOLAP)*. ACM, 2015: 75–84.
- A. Löser, [S. Arnold](#), and T. Fiehn. The GoOLAP Fact Retrieval Framework. *Business Intelligence*. Springer, 2012: 84–97.