Wiki2Prop: A Multimodal Approach for Predicting Wikidata Properties from Wikipedia

Michael Luggen
University of Fribourg
Fribourg, Switzerland
michael.luggen@unifr.ch

Julien Audiffren
University of Fribourg
Fribourg, Switzerland
julien.audiffren@unifr.ch

Djellel Difallah
New York University Abu Dhabi
Abu Dhabi, UAE
djellel@nyu.edu

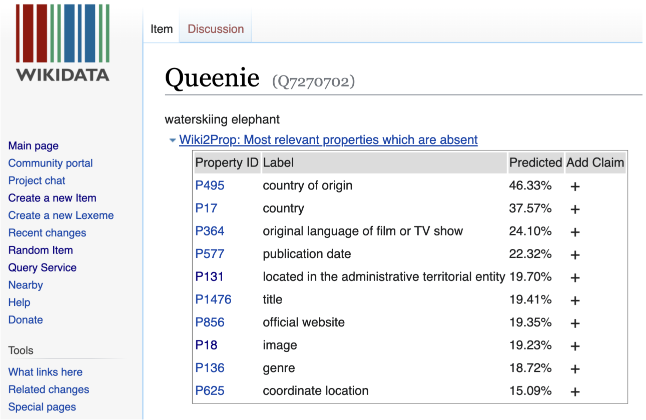
Philippe Cudré-Mauroux
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

ABSTRACT

Wikidata is rapidly emerging as a key resource for a multitude of online tasks such as Speech Recognition, Entity Linking, Question Answering, or Semantic Search. The value of Wikidata is directly linked to the rich information associated with each entity – that is, the properties describing each entity as well as the relationships to other entities. Despite the tremendous manual and automatic efforts the community invested in the Wikidata project, the growing number of entities (now more than 100 million) presents multiple challenges in terms of knowledge gaps in the graph that are hard to track. To help guide the community in filling the gaps in Wikidata, we propose to identify and rank the properties that an entity might be missing. In this work, we focus on entities with a dedicated Wikipedia page in any language to make predictions directly based on textual content. We show that this problem can be formulated as a multi-label classification problem where every property defined in Wikidata is a potential label. Our main contribution, Wiki2Prop, solves this problem using a multimodal Deep Learning method to predict which properties should be attached to a given entity, using its Wikipedia page embeddings. Moreover, Wiki2Prop is able to incorporate additional features in the form of multilingual embeddings and multimodal data such as images whenever available. We empirically evaluate our approach against the state of the art and show how Wiki2Prop significantly outperforms its competitors for the task of property prediction in Wikidata, and how the use of multilingual and multimodal data improves the results further. Finally, we make Wiki2Prop available as a property recommender system that can be activated and used directly in the context of a Wikidata entity page.

CCS CONCEPTS

• **Human-centered computing** → *Wikis*; • **Information systems** → **Incomplete data**; *Network data models*; • **Computing methodologies** → *Supervised learning*.



Queenie (Q7270702)

waterskiing elephant

Wiki2Prop: Most relevant properties which are absent

Property ID	Label	Predicted	Add Claim
P495	country of origin	46.33%	+
P17	country	37.57%	+
P364	original language of film or TV show	24.10%	+
P577	publication date	22.32%	+
P131	located in the administrative territorial entity	19.70%	+
P1476	title	19.41%	+
P856	official website	19.35%	+
P18	image	19.23%	+
P136	genre	18.72%	+
P625	coordinate location	15.09%	+

Figure 1: Predictions of properties by Wiki2Prop on the Wikidata entity Q7270702. With a *Wikidata gadget*, the relevant new properties can be presented directly on the page of the respective Wikidata entry to support editors with completing the entry.

KEYWORDS

Multimodal Deep Learning, Knowledge Graph, Property Prediction, Wikipedia, Wikidata

ACM Reference Format:
Michael Luggen, Julien Audiffren, Djellel Difallah, and Philippe Cudré-Mauroux. 2021. Wiki2Prop: A Multimodal Approach for Predicting Wikidata Properties from Wikipedia. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3450082

1 INTRODUCTION

Knowledge graphs are used to improve a multitude of online tasks such as Speech Recognition, Entity Linking, Question Answering, or Entity-centric Search. Wikidata, in particular, is a free collaborative knowledge graph maintained by a community of editors who take great care in manually curating the resource, and by a set of automated agents (a.k.a. bots) that extract properties relating to entities from Wikipedia and external sources. The bulk of the bots collect their information from (semi-)structured data contained

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450082

in Wikipedia, e.g., Infoboxes, Tables, Lists, and Categories. Consequently, popular entities that have an extensive Wikipedia page have been enriched using a number of semi-automated methods and are therefore well-established in terms of their properties on Wikidata. Conversely, less popular entities that have Wikipedia articles with little to no structured content, do not benefit from the application of automated structured information extraction methods. For instance, Queenie (the water-skiing elephant) Wikipedia page[1] contains at the time of writing no structured information.

The distribution of property count among Wikidata entities follows a heavy-tailed distribution, with an average of 3.93 different properties per entity; This observation also applies to those entities that have corresponding Wikipedia pages across multiple language editions, with a slightly higher average of 4.96 property per entity [12]. The problem is compounded for entities that belong to rare classes. For instance, Queenie belongs to classes *mammal living in captivity* and *Asian elephant*, which are relatively under-represented in Wikidata. Conversely, Wikipedia articles tend to include a larger amount of unstructured information – as well as multimedia elements, such as images. For instance, the article about Queenie describes a fair amount of facts, but these are not structured in any way (e.g., it contains no Infobox, which is typical for long-tail entities [40]). Furthermore, the article also includes a picture of the elephant – an information-rich element that is currently not exploited by these automated methods. Actually, most of Queenie's[2] properties were manually entered by a Wikidata Editor, including *place of birth*, *sex or gender*, *occupation*, *date of birth* and *date of death*, which were all available in the Wikipedia article, although in an unstructured form.

In this paper, we propose a new method that can propose missing properties, whose values can later be filled by human editors or bots, for a given entity by analysing the unstructured textual content of Wikipedia. Formally, our method, coined *Wiki2Prop*, tackles the problem of correctly predicting relevant properties of an input entity through a multimodal approach that leverages its related Wikipedia article. One key ingredient of our approach is to consider properties as *labels*, and to model the problem as an instance of multi-label learning with incomplete class assignments [5]. In this setting, each entity is associated with one or more feature vectors, which are assumed to contain information related to the entities (unknown) collection of potential properties. Wiki2Prop learns a model to generate a probability for each label defined in Wikidata. As such, once the labels of an input entity are ranked using the output probabilities of our model, the editors (or bots) can focus on the more incomplete entities and the more promising properties first in order to fill out missing information on Wikidata efficiently and effectively. Additionally, an entity might have varying amounts of complementary information present across Wikipedia pages and languages, or in non-textual format, such as images. One main advantage of our proposed architecture is that it can work with an arbitrary number of available input from non-english languages, i.e., our method is both able to perform well with English-only embeddings while being able to seamlessly integrate features from additional languages (e.g., German and French embeddings), when available, to achieve even better performance. Moreover, Wiki2Prop can also extract information contained into the entity *image of interest* to improve its predictions – as it has been shown that images are a key source of additional information (see e.g. [13, 39]).

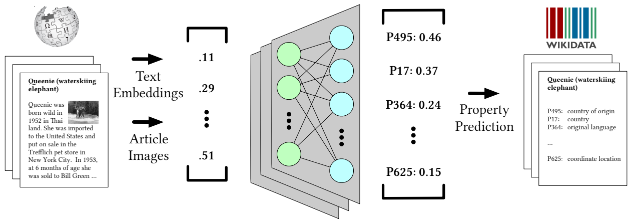

Queenie (waterskiing elephant)
Queenie was born wild in 1952 in Thailand. She was imported to the United States and put on sale in the Trefflich pet store in New York City. In 1953, at 6 months of age she was sold to Bill Green ...
Text Embeddings
Article Images
.11
.29
.51
P495: 0.46
P17: 0.37
P364: 0.24
P625: 0.15
Property Prediction
WIKIDATA
Queenie (waterskiing elephant)
P495: country of origin
P17: country
P364: original language
...
P625: coordinate location

Figure 2: A high level illustration of the Wiki2Prop architecture. First, it extracts information from unstructured data contained in Wikipedia articles using a carefully tuned Word2Vec architecture. The resulting embedding of a given entity is combined with images included in the article (when available) and then used as input to a Neural Network trained to predict the relevant properties.

To summarize, we introduce the following contributions:

1. We tackle the question of property recommendation in large scale collaborative knowledge base construction. Unlike existing approaches, our method does not rely on hard-coded rule-sets or heuristics, instead it performs information extraction from unstructured content related to entities of interest. The properties predicted by Wiki2Prop can be used to enrich Wikidata (See an example of the functionality in Figure 1).
2. We introduce a new Multi Modal Neural Network architecture that is able to combine the embedding information originating from Wikipedia entries in multiple languages – namely English, German and French (EN, DE, and FR for short) – and non textual data (images) to produce a confidence vector and predict relevant properties. Furthermore, we learn directly from a large set of existing Wikidata instances and do not require a manually labeled dataset.
3. We report on an extensive performance evaluation campaign of our method on real Wikidata and Wikipedia data, and show that it consistently outperforms the state of the art across a wide range of metrics that measure properties predictions, ranking and coverage.
4. We build a property recommender system based on Wiki2Prop, and deploy it on the Wikimedia infrastructure. Wiki2Prop recommendations are available through an API, and is integrated into Wikidata interface (as a Gadget) for the Wikidata users who enable it.

The rest of this paper is organised as follows. In Section 2, we briefly present the state of research in predicting Wikidata properties. We also discuss the state of the art in entity embeddings, multi-label learning and multi methods, whose fields Wiki2Prop is at the intersection of. In Section 3, we detail the inner working of each part of Wiki2Prop. Finally, in Section 4 we extensively evaluate Wiki2Prop using a large collection of metrics before concluding.

[1] https://en.wikipedia.org/wiki/Queenie_(waterskiing_elephant)
[2] https://www.wikidata.org/wiki/Q7270702

2 RELATED WORK

Word and Entity Embeddings. Word2Vec[21] is an established method to embed unstructured text in a multidimensional vector space that represents inherent semantic concepts. Since its inception, multiple specialized embeddings based on Word2Vec where developed. As input for our methods, we use the Wikipedia2Vec framework described in [42]. Wikipedia2Vec combines three submodels to learn embeddings from Wikipedia. The text of each article is embedded with a word-based skip-gram model. This model is combined with a Link Graph model that embeds entities that are neighbors in the link graph. Finally, an anchor context model aims to place the entity and word embeddings close to each other in the vector space. It does so by embedding the words adjacent to Wikipedia's internal links.

Image Embeddings. Deep Neural Network architecture have been shown to be extremely effective at extracting information from images [33]. One of the most popular network to achieve this is Densenet, who has shown good results on challenging image classification tasks [16]. In particular, multiple previous works have successfully used pretrained versions of Densenet for a large variety of image related tasks, such as early tumor detection [6, 37]. Following these promising results, we used the pretrained Densenet-121 network [16, 30] to extract information from the Wikipedia images. Image embeddings are used to situate and relate images as first class citizens in a knowledge graph [26] and also to build *same as* relationships[18] between different knowledge graphs. Both works report about a additional expressive power which aligned images bring to knowledge graph entries.

Property Recommender Systems. Wikidata provides an API[3] for suggesting missing properties. The approach used by this property recommender is based on Association Rules, inspired by [1], but enhanced with contextual information like the class membership of the entity for which properties are suggested. In [46], a formal evaluation comparing state-of-the-art recommender systems for this task is provided, including [14], where the authors improved previous results by using a tree-based method. However, to the best of our knowledge, Wiki2Prop is the first to use external information (Wikipedia) to enrich Wikidata. In [34], the task of property ranking in Wikidata is discussed. To compare the different property ranking approaches, a crowdsourced ground-truth dataset is created by comparing properties pair-wise in regard to their importance in the context of a single entity. While this dataset can be used to evaluate the property ranking task, it is unfortunately not suitable for the property prediction task we tackle. This is due the low number of only pair-wise comparisons in this ground-truth dataset. Recoin [2] is a method that provides a ranking of missing properties based on the probability that a property appears in the class the entity belongs to. Furthermore, the method classifies the entities in regards to completeness. This method directly depends on the correct classification of the entity it is applied to (knowing that roughly 7% of the entities in Wikidata also present in Wikipedia do not belong to any class). Also, the method works best on classes that use properties uniformly. To counteract this, Recoin applies a set of heuristics, namely on the class of *Humans* (q5), to introduce more fine-grained classes. We formally evaluate Recoin and compare it to our proposed method in Section 4. In [31], meta-information on the completeness of Wikidata are discussed. The method focuses on evaluating the completeness of entities per property (e.g., are all cantons (states) of Switzerland available through the statement "contains administrative territorial entity."?). At this point, we would also like to point to two meta-studies that embed this work in a broader context of research conducted on Wikidata. [11] provide an overview of the research performed on Wikidata through a systematic study, where the authors identify current research topics as well as research gaps that require further attention. [29] specifically tackles data quality in Wikidata. The paper surveys prior literature and show that a number of quality dimensions (e.g., accuracy and trustworthiness) have not been adequately covered yet.

Other Knowledge Gap Tasks. Incomplete data in knowledge graphs is a severe problem, especially in collaborative setups, where human editors spend a significant amount of time identifying and fixing the gaps in the knowledge base. Several tasks have been defined and tackled in the literature. For example, *predicting entity types* aims at recommending a type for new instances, Moon et al. [22], propose to learn embeddings for entity types. Luggen et al. [19] proposed to tackle the task of identifying incomplete classes in a knowledge graph by leveraging editor activity patterns. Another line of work focuses on *link prediction* or identifying the relationship between two input entities. Here, a series of works proposed advanced methods to build joint entity and relation embeddings (see e.g., [17] and references therein). Knowledge graph integration aims at connecting repositories in Linked Open Data (LOD) to obtain complementary information about the same entity in different databases [9]. Wikipedia articles in conjunction with Wikidata properties are further used in a multi-class document classification to discover semantic relations between Wikipedia Articles [28]. Another knowledge gap related task is *entity mapping* which aims at identifying snippets of text in Wikipedia discussing a specific entity [27]; this information has the potential to help build robust joint entity-text embeddings for tail entities and thus improve property prediction.

Multi-Label Classification. The Multi-Label Classification framework, also called Multi-Label Learning (MLL), is hardly new, and has been linked to text categorization since its inception (see e.g., [36]). Multiple approaches specifically designed for MLL have been proposed, and we give below a brief overview of recent contributions to this field below, focusing on neural network-based approaches (we refer the reader to e.g., [23] for a complete review of other methods). In the seminal work of [47], the authors introduced a new MLL tuned loss function, called BPMLL (Backpropagation for Multi-Label Learning), and showed that even simple networks using this loss function yield good performance on text categorization tasks. Similarly, [24] showed that on large-scale datasets, their properly tuned neural network NN_{AD} with one hidden layer achieved state-of-the-art performance. More recently, [50] proposed a method that first uses a Deep Learning approach to learning a proper embedding of the underlying labels network , and then uses the resulting manifold to train a k-Nearest Neighbours algorithm. This method can capture complex networks of label dependencies, but cannot be

[3] https://www.wikidata.org/w/api.php?action=help&modules=wbsgetsuggestions

easily applied to predicting properties on Wikidata, as A) it suffers from the disproportionate influence of rare labels over vertices and B) its shares some of the k-Nearest Neighbours weaknesses, such as computationally expensive predictions on very large datasets.

Multi-Modal Learning. In recent years, Multi-Modal Learning – and in particular the fusion of information originating from different sources – has received increased attention from the research community, as it been shown that this approach can lead to significantly improved performance in real world scenarios [3, 13]. [45] used a tensor product of all the embedded modalities, while [38] used Gaussian restricted Boltzmann machines to model the relation between the modalities. More recently, [35] used text-only multimodal techniques to show that the combination of different word embeddings can improve the performance of sentiment analysis on short texts, and [41] proposed a multimodal approach that combines relationship t riplets and images to improve knowledge representation. However, to the best of our knowledge, Wiki2Prop is the first method that uses multi language embeddings and images. Additionally, Wiki2Prop is able to account for missing modalities, a task that is still challenging the Multi-Modal Learning community. Indeed, most previous contributions in these domains try to infer missing modalities before prediction (see e.g., [7, 15] and references therein), which leads to significant error compounding. Conversely, Wiki2Prop fusing layer (see Section 3.3.2) directly merges available modalities and avoids the negative impact of missing modalities on both training and prediction tasks by transferring additional information from the input to the fusion layer using Dirac δ functions.

3 METHOD

This section introduces and details the inner working of Wiki2Prop, as summarized in Figure 2. We start by discussing the embedding of the unstructured data of Wikipedia entities in multiple languages in Section 3.1. In Section 3.2, we briefly recall the Multi-Label Learning framework, while we detail the architecture of the neural network part of Wiki2Prop in Section 3.3.

3.1 Data processing

As aforementioned, Wiki2Prop extracts unstructured information from the English, German, and French versions of Wikipedia articles, as well as included images. Our model requires that a given entity from Wikidata has a corresponding English Wikipedia article to predict its potential properties. The model can integrate multiple additional language resources as well as images. However, we limit our experiments to the German and French wikis since, together with English, these capture the largest overlap with Wikidata.

Wikidata Structure Primer. Entities in Wikidata are complex elements that possess a unique *Identifier* (an integer prefixed with Q), *Labels*, *Descriptions* and *Aliases* as meta information, the latter three in multiple Languages. Every entity further owns multiple *Statements* which are combined by a *Property*, and *Literal* or a *Link* to another Entity. Each of the Statements posses further *Qualifiers* (e.g., time period of validity) and *References* (a link to an external source of the statement). Finally, each entity may possess *Sitelinks*, which are links to other Wikimedia projects, foremost to multiple languages versions of Wikipedia articles, constituting a high-quality mapping between Wikipedia and Wikidata for all languages.

Embedding Wikipedia with Wikipedia2Vec. To extract unstructured information from Wikipedia articles, we proceed as follows. First, we use Wikipedia2Vec[4] [42] to train an embedding on the complete English, French, and German Wikipedia dumps. Wikipedia2Vec uses an extension of the Skip-gram model proposed by [43] that jointly optimizes a Wikipedia link graph model, a word-based skip-gram model, and an anchor context model. We compute multiple embeddings of the Wikipedia dumps using varying parameters. We used the default parameters of the embedding framework, except for

- the *window size* was set to 10 for the skip gram model which improves the performance on all methods;
- the *min entity count* was set to 1 to enable the inclusion of all entities in the embedding regardless of their occurrence count.

After an initial evaluation on the number of dimensions we choose to embed with 300 dimensions, this is also a frequent choice for word embeddings (see [4]). Additionally, we evaluated the influence of the Wikipedia link graph model component in order to estimate the importance of the Wikipedia network on our task. Our tests show that the use of the graph model in the embeddings consistently improves the prediction performance on the applicable baselines and our proposed model.

Extracting Wikimedia Commons Images. We downloaded a representative image per article from Wikimedia Commons, which is the sister-project of Wikipedia storing the media assets. For each entity, we used the image linked from its Wikidata property *P18: image* [5] (when available). The property P18 is described as: "image of relevant illustration of the subject". For the minority (less than one percent) of entries which have more than one image attached, we choose the first image linked from the entry. In total, a $n = 1'156'380$ images have been collected. Similar to [16] we prepare the images as follows, the images were resized to a width of 224 pixels (if height > width) or a height of 224 pixels otherwise, before being croped to a size 224 × 224.

Extracting the properties. To extract the labels y associated with each entity x, we proceed as follows. First, we extract all the properties of each entity of Wikidata that have an EN representation in Wikipedia (more specifically an *enwiki sitelink*), for a total of $n = 7'768'807$ entities. Then, these pieces of data are filtered using the following set of rules:

- we extract only distinct properties, regardless of the number of times they are attached to a single entity.
- all Qualifiers and Reference information attached to the statements are discarded, as they are not part of our model;
- each Property of the type *external links* is dismissed, as those properties can generally be efficiently populated by importing[6] from the target databases. This specific rule reduces

[4] https://wikipedia2vec.github.io/
[5] https://www.wikidata.org/wiki/Property:P18
[6] https://www.wikidata.org/wiki/Wikidata:Dataset_Imports

the number of possible properties from more than six thousands to less than two thousands, significantly improving the learning process and generalization of Wiki2Prop.

Joining the final dataset. Finally, we generate the final dataset by matching the collected properties with their corresponding entity vectors from the embedding, in each of the available languages, as well as the resized image. When an entity has an EN embedding but is missing DE and FR embeddings or has no image, those missing representations are replaced with 0-values vectors during the training/testing of Wiki2Prop.

To further improve the performance of property predictions, we removed all properties that are currently extremely rare (i.e., that occur less than 100 times in the entirety of Wikidata). We note that extremely rare properties cannot be learned effectively, as they occur in less than 0.02% of the entities, and can hinder the learning process of more popular properties by creating large gradient deviations in optimizers such as AdaGrad. Following this property pruning, we remove from the dataset entities which do not own any properties anymore, resulting in $n = 4'680'290$ entities featuring $k = 991$ distinct properties. For the empirical evaluation of our result (Section 4), the resulting dataset is then split into a training set (70%) and a test set (30%). The training set is used to train Wiki2Prop, as well as other methods that are compared to it, while the test set is used to evaluate the performance of the different methods.

We source the datasets of Wikipedia[7] (for the respective languages *en*, *de*, *fr*) and Wikidata[8] from a dump. The images were downloaded individually from Wikimedia Commons[9]. We provide the source code[10] build for the performance evaluation of Wiki2Prop. The above described data processing steps are resource intensive. Therefore we also provide intermediate snapshot steps of the datasets used for the evaluation.

3.2 Multi-Label Learning

In this work, the property prediction task is modeled as a Multi-Label Learning problem. Formally, this framework is defined as follows: let $\mathcal{X}$ a Polish space (here $\mathcal{X} = \mathbb{R}^d$) and $\mathcal{Y} = \{0,1\}^k$ be respectively the *entities* and the *label* spaces. The label space encodes the presence or absence of k knowledge graph properties for a given entity. Let $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ be a dataset of N pairs of an entity-labels (x, y). In the following, we denote by y_i the value of the i-th label: $y_i = 1$ if and only if the label i is present (i.e., the entity has the k-th property). The objective is to learn a function $\ell : \mathcal{X} \mapsto \mathcal{Y}$ that is able to predict the labels of entities, i.e., $\ell(x) \approx y$, even for an entity-label pair (x, y) unseen in $\mathcal{D}$.

While this problem can easily be rewritten as a multi-class classification problem, by considering each element of $\mathcal{Y}$ – i.e., each possible combination of labels – as a different class, this approach leads to an exponential number of classes, which quickly becomes intractable (approximately $2^{991} \approx 10^{300}$ classes in the Wikidata dataset). Instead, previous works in multi-label learning (such as [47]) have focused on learning a *confidence* function

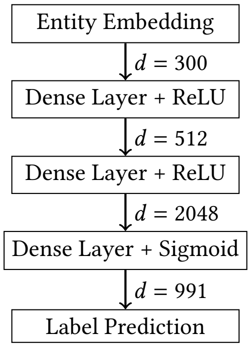


Figure 3: Architecture of the Language specific networks. d indicates the dimension of the output of each layer, while the activation function of each layer is written with a prefix "+". All three networks (EN, DE, and FR) share the same architecture, but are initialized randomly and trained on different training sets. The image embedding network was based on a different architecture (Densenet 121).

$$\begin{aligned} h: \quad & \mathcal{X} \mapsto [0,1]^k \\ & x \to h(x) = (h(x)_1, \dots h(x)_k) \end{aligned}$$

where $0 \leq h(x)_i \leq 1$ is the confidence that x has the label i. Intuitively, a confidence function h is deemed good if $h(x)_i$ is close to 1 if i is a true label, and is close to 0 otherwise. This function naturally induces for each entity x a ranking of labels, where the label i is deemed more likely than the label j if $h(x)_i > h(x)_j$. This interpretation is particularly useful when evaluating the predictions (see Section 4.2). A label prediction function can be derived from h by choosing a threshold $\tau \in [0,1]$, and defining

$$h_\tau(x) = \left(\mathbb{1}\left(h(x)_1 > \tau\right), \dots, \mathbb{1}\left(h(x)_K > \tau\right)\right),$$

where $\mathbb{1}$ denotes the indicator function. Note that the proper choice of τ is key to obtain good precision/recall performance (see Section 4). Following previous works (see [10, 24]) we use a threshold $\tau(x, i)$ that is both label and entity-dependent.

3.3 Multimodal Multi-Label Learning: Wiki2Prop

While several previous works have used Neural Network-based algorithms for Multi-Label Learning (see [50] and references therein), Wiki2Prop introduces a new multimodal multi-language approach to the problem that is robust to incomplete data, i.e., data points for which at least one language is missing. More precisely, the multimodal architecture developed in Section 3.3.2 takes advantage of the additional information contained in Wikipedia entries in other languages as well as images when they are available while being able to predict labels from the English embedding alone if other data are missing.

To obtain this flexibility, Wiki2Prop requires a multi-step training that first trains the Neural Network architecture to use each language embedding separately all four layers from top to bottom and then in a second step trains the final Wiki2Prop model

[7] https://dumps.wikimedia.org/{language}wiki/20180901/
[8] https://dumps.wikimedia.org/wikidatawiki/entities/20180813/
[9] https://commons.wikimedia.org/
[10] https://github.com/eXascaleInfolab/Wiki2Prop/

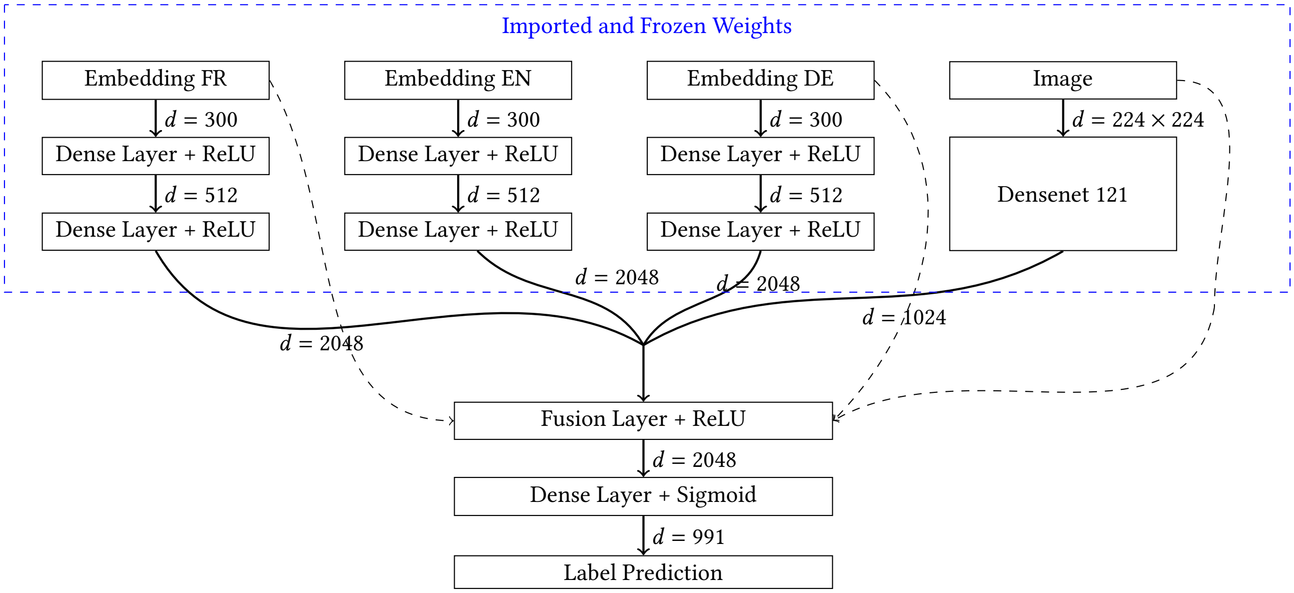


Figure 4: Wiki2Prop architecture. d indicates the dimension of the output of each layer, and the activation function of each layer is written with a prefix "+". The dashed lines represent the Dirac information transfer used by the fusion layer. The dashed blue rectangle surrounds the weights that are initialized using the previously trained networks, and then frozen. Note that the Fusion layer is detailed in Section 3.3.2

to combine the available multimodal embeddings. The first step in training Wiki2Prop, whose architecture is detailed in Figure 4, relies on training smaller language-specific networks.

3.3.1 Language Specific embedding. For each chosen language (EN, DE, and FR), we train a dedicated Neural Network to predict the entity labels y using the available (i.e., non-zero) language embedding (resp. x^{EN}, x^{DE} and x^{FR}). Similarly to [24], we only use fully connected layers. The network uses three consecutive dense layers, with either Rectified Linear Units (ReLU) [8] or Sigmoid as activation functions (see Figure 3 for the details of the network's architecture). Before the training, each layer weights and biases are randomly initialized using the uniform distribution over $[-1, 1]$.

Training process. The network is trained using the subset of the training set that has the corresponding language embedding; consequently, the size of the training set is different for each language. We use the stochastic gradient descent algorithm with Nesterov momentum [25] to optimize the $\mathcal{L}^2$-regularized Binary Cross-Entropy loss.

$$\begin{aligned} \mathrm{BCE}(y, h(x)) = &\sum_{i=1}^{K} (1 - y_i) \log (1 - h(x)_i) \\ &+ \sum_{i=1}^{K} y_i \log (h(x)_i) + \lambda \|W\|_2^2 \end{aligned} \tag{1}$$

where (x, y) is an entity label pair in the training set, $h(x)$ is the output of the network for x, W is the vector containing all the network weights and biases, and λ is the regularization parameter. Each network was trained using Early Stopping [32], and their hyperparameters were optimized using a 10-fold cross validation, resulting in a learning rate $\eta = 0.1$, and $\lambda = 10^{-5}$.

3.3.2 Seamless Multimodal Fusion. The Wiki2Prop architecture – detailed in Figure 4 – is composed of two successive groups of layers. The first group includes the first two layers of each previously trained language-specific network, as well as the pretrained Densenet 121 network [16] for the image embedding. Each language specific network outputs an embedding of their respective Wikipedia page of dimension $d = 2048$, and Densenet produces a image embedding of dimension $d = 1024$. The second group includes a Fusion layer that combined all the outputs of the first groups (see below), and one additional dense layer. For the second part of the training, the weights of each layers of the first group are initialized using their pretrained weights (see Section 3.3.1). These weights are then frozen, following neural network transfer learning techniques [44]. The weights of the second group are randomly initialized using the uniform distribution over $[-1, 1]$. The idea behind this two-step training process is that through the first step, the first group of layers are trained to extract features from each of the language and from images separately, while in the second step, Wiki2Prop learns to combine these features to achieve a better prediction of the entity property.

Fusion Layer. The Fusion Layer combines the outputs of the previous layers as follows:

$$\begin{aligned} &\mathrm{FUSION}\left(o^{EN}, o^{DE}, o^{FR}, o^{IMG,} \delta^{EN}, \delta^{DE}, \delta^{FR}, \delta^{IMG}\right) \\ &\quad = \delta^{EN} W^{EN} o^{EN} \\ &\quad + \delta^{DE} W^{DE} o^{DE} \\ &\quad + \delta^{FR} W^{FR} o^{FR} \\ &\quad + \delta^{IMG} W^{IMG} o^{IMG} + b_{\mathrm{FUSION}} \end{aligned}$$

where

- o^{EN} (resp. o^{DE}, o^{FR},o^{IMG}) is the output of dense layers that follows the EN embedding (resp. DE and FR, and the second to last layer of Densenet 121),
- δ^{EN} (resp. δ^{DE}, δ^{FR}, δ^{IMG}) $\in \{0,1\}$ is the Dirac delta function that is equal to 1 if this entity has a EN embedding (resp. DE, FR, and a image) and 0 otherwise,
- $W^{EN}, W^{DE}, W^{FR}, W^{IMG}$, (resp. b_{FUSION}) are weights matrices (resp. bias vector) updated during back-propagation.

The main difference between the Fusion layer and a Dense layer applied to the concatenation of o^{EN}, o^{DE}, o^{FR} and o^{IMG} are the Dirac delta functions. They ensure that if, for instance, the FR embedding is missing, the corresponding weight matrix W^{FR} is not used for prediction neither updated during back-propagation (since the gradient is canceled). This effect is key for the following reason. First, note that $x^{FR} = 0$ does **not** imply $o^{FR} = 0$, due to the bias of the first dense layer. Henceforth, without δ^{FR}, W^{FR} would be updated at each back-propagation, even when the FR embedding is uninformative. Since entities with FR embeddings represent only a fraction of the total entities ($1'185'590$ of $7'768'807$), W^{FR} would be heavily biased by uninformative updates. In summary, the use of δ^{FR} implies that W^{FR} is only used and updated when FR embeddings are available, and is not penalizing the network when only EN embeddings are present.

Training process. Before the training of Wiki2Prop, the training set is augmented as follows: for each entity, each combination of available embeddings and modes that includes EN is added to the training set. For instance, if EN and DE embeddings are available, then both EN & DE and EN only are added to the training set. This augmentation step allows Wiki2Prop to learn to predict labels even when only EN embeddings are available, while still learning to take advantage of the additional information. Similarly to Section 3.3.1, we use the stochastic gradient descent algorithm with Nesterov momentum to optimize the $\mathcal{L}^2$-regularized Binary Cross-Entropy loss (1), using Early Stopping. The network hyperparameters were optimized using a 10-fold cross validation, resulting in a learning rate of $\eta = 0.1$, and $\lambda = 10^{-7}$.

4 EVALUATION

In this section, we perform an extensive experimental analysis of Wiki2Prop and highlight the improvement it can bring to Wikidata property prediction. We first produce an ablation analysis of our model, and show that the inclusion of multiple languages and images increase its performance. We compare the performance of Wiki2Prop to a state-of-the-art method used for Wikidata completion, namely Recoin [2], as well as recent contributions in Multi-Label predictions (BPMLL [47] and NN_{AD} [24]). The results show that Wiki2Prop performs significantly better than all the other methods.

4.1 Datasets

To train and evaluate our model and baselines, we utilize the existing Wikidata (keeping entities having a Wikipedia sitelink as described in section 3.1) and their current set of properties. Since Wikidata is by default *incomplete*, properties that would apply to some entities may be missing. Moreover, it would be unreasonable to manually build a ground truth, as the total number of entities of Wikidata is too large and the set of properties virtually unbounded.

As mentioned in section 3, our dataset is composed of $\approx 5M$ entities. Given the large size of this dataset, we use a holdout evaluation by splitting it into a training set (70%) and a test set (30%). Note that the training set is used both to train Wiki2Prop and the other methods that are compared to it, and to select hyperparameters using cross validation. In addition, to analyse the behaviour of the methods in regard to their existing property count per entity, which is a proxy for how complete an entity is, we create the subsets from the test set of entites with at least 1, 5, 10, 15, 20 and 25 properties respectively.

4.2 Evaluation Metrics

One of the most challenging aspects of the multi-label learning setting is the proper evaluation of the results. Contrary to a regular classification problem, a large range of metrics has been designed to evaluate different aspects of the results (see e.g., [49]). To assess the performance of the different methods, we use the two following collections of metrics, one to assess the quality of label prediction, and one to evaluate the quality of the ranking estimation.

- *Label Prediction Evaluation.* These metrics directly assess the quality of label prediction $h_\tau(x)$. We included the commonly used Micro version of Recall, Precision and F1 (see [48] for an extended definition of these metrics in the Multi Label prediction setting). We also used the Hamming Loss – defined as the number of modifications necessary to transform the predicted sequence of labels into the true sequence of labels y [48].
- *Ranking Evaluation.* These metrics evaluate the ranking of labels induced by the confidence vector $h(x)$, where the label are ranked by decreasing order of confidence (and thus, the label with the highest confidence is ranked first). The idea behind these metrics is that true labels should have a better ranking that false labels. We included the Instance-AUC [5] – where the predicted label scores for each entity are ranked, and the resulting AUCs is averaged over all the entities. We also used Mean Reciprocal Rank (MRR) [20], which computes the average of the inverse of the rank of true labels. Finally, we chose to include the Coverage metric [36], which is defined as the rank of the last true label, i.e. the relevant property that was given the smallest amount of confidence.

4.3 Baselines

In our experiments, we compare the performance of Wiki2Prop to five different methods, including Wiki2Prop without images, and Wiki2Prop restricted to english only – to highlight the flexibility of our algorithm with respect to additional langages as well as the benefits they bring. The other methods are Recoin [2], the current state-of-the-art method for property prediction in Wikidata, BPMLL [47] and NN_{AD} [24], two recent contributions from the multi label prediction community.

	$P_\geq$	W2P	W2P_txt	W2P_En	NN_AD	BPMLL	Recoin
Hamming Loss ↓	1	**2.601**	2.632	2.969	3.667	9.958	3.394
	5	**3.051**	3.083	3.488	4.165	10.28	4.124
	10	**3.836**	3.936	4.541	5.234	11.50	5.911
	15	**5.130**	5.131	6.143	6.907	15.51	8.783
	20	**6.645**	6.652	8.111	9.025	21.96	11.90
	25	**8.790**	8.876	10.90	12.12	28.49	15.94
Micro - F1 ↑	1	**0.790**	0.764	0.730	0.686	0.296	0.726
	5	**0.828**	0.790	0.757	0.723	0.356	0.699
	10	**0.851**	0.817	0.783	0.758	0.400	0.743
	15	**0.853**	0.832	0.790	0.770	0.335	0.698
	20	**0.849**	0.838	0.794	0.775	0.2200	0.675
	25	**0.838**	0.831	0.781	0.760	0.148	0.640
Micro - Precision ↑	1	**0.795**	0.788	0.765	0.675	0.249	0.748
	5	**0.866**	0.838	0.822	0.746	0.346	0.844
	10	**0.896**	0.875	0.867	0.809	0.498	0.891
	15	**0.904**	0.890	0.886	0.840	0.552	0.902
	20	0.899	0.890	0.887	0.847	0.482	**0.938**
	25	0.884	0.880	0.879	0.842	0.429	**0.946**
Micro - Recall ↑	1	**0.784**	0.740	0.698	0.697	0.365	0.705
	5	**0.794**	0.747	0.702	0.701	0.366	0.699
	10	**0.810**	0.767	0.714	0.713	0.334	0.633
	15	**0.807**	0.780	0.713	0.710	0.240	0.561
	20	**0.805**	0.792	0.719	0.714	0.143	0.527
	25	**0.796**	0.787	0.703	0.692	0.089	0.484
Instance-AUC ↑	1	**0.998**	**0.998**	0.997	0.995	0.962	0.961
	5	**0.998**	**0.998**	0.997	0.996	0.961	0.990
	10	**0.998**	**0.998**	0.997	0.996	0.958	0.992
	15	**0.998**	**0.998**	0.997	0.995	0.954	0.991
	20	**0.998**	0.997	0.996	0.994	0.919	0.989
	25	**0.997**	0.996	0.994	0.991	0.895	0.989
MRR ↑	1	**0.950**	0.850	0.833	0.798	0.395	0.810
	5	**0.997**	0.964	0.952	0.926	0.538	0.954
	10	**0.999**	0.989	0.982	0.967	0.715	0.988
	15	**1.000**	0.996	0.990	0.980	0.688	0.989
	20	**1.000**	0.999	0.994	0.982	0.536	0.986
	25	**1.000**	0.998	0.993	0.983	0.440	0.978
Coverage ↓	1	**14.7**	**14.7**	17.7	22.0	109.3	28.1
	5	**16.9**	19.2	20.7	26.1	138.5	32.4
	10	**23.7**	26.4	29.4	36.7	183.9	40.1
	15	**33.7**	37.2	42.6	53.7	263.2	55.2
	20	**46.4**	51.0	59.7	77.1	366.5	78.9
	25	**63.5**	70.2	84.5	113.3	481.3	88.7

Table 1: The performance of W2P (Wiki2Prop full model with text and images) compared to W2P_txt (Wiki2Prop multilingual text only and without images), W2P_En (Wiki2Prop with English text only and without images), Recoin, NN_AD and BPMLL, on the test set ($n = 1'404'090$), and subsets with entities featuring at least a minimum number of properties ($P_\geq$). The best scores are written in bold. A MRR of 1.0 means that the property predicted with the highest confidence by Wiki2Prop was almost always correct.

4.4 Performance of Wiki2Prop

In Table 1, we compare the baseline methods against our proposed method Wiki2Prop (W2P) on the complete test set consisting of 1'404'090 entities. Besides the baseline methods, and for ablation analysis, we also report a version of our method trained without images, but with all additional languages (W2P_txt, for Text-Only), as well as a version trained solely with the English Wikipedia embeddings and no images (W2P_En). Each column represents a method, each row represents a metric, where the sub-rows $P_\geq$ (from 1 to 25) represent the resp. sub-sets of entities which have at least the shown amount of properties.

Instance-AUC. First, note that all methods tested in our experiments achieved very high instance-AUC. This is due to the fact that only a handful of properties are relevant for each entity, and all methods were able to easily eliminate the major part of the irrelevant properties (out of 987 candidates properties), resulting in a overall high AUC. Yet, subtle but crucial improvement can be seen at the top of the ranking between the different algorithms. This difference is further explored by the other metrics, such as Coverage.

Ablation Analysis. First, it is interesting to note that Wiki2Prop also improves on all metrics when provided with the embeddings of the additional languages and images compared to W2P_txt (i.e. without images). Moreover, W2P_txt in turn outperforms the model trained with only the English embedding (W2P_En). This improvement is consistent regardless of the number of properties P of the entities. This shows that both the inclusion of images, and the multi-language approach improve the performance of W2P, even when the number of available languages – or the presence of relevant images – is not consistent across entities.

Comparison with Baselines. Wiki2Prop outperforms the baselines on all metrics (except for Recoin in one particular case discussed below), highlighting the advantages of using Wikipedia embeddings and multimedia data to learn which of the properties can be attached to an entity. Importantly, the advantage of Wiki2Prop is particularly visible for the Micro-Recall and Micro-Precision metrics. This is key as these values represents the capacity of a method to predict most of the missing property (Micro-Recall) correctly (Micro-Precision). Note that the difference between Wiki2Prop Micro Recall and other methods increases significantly with P (the minimum number of properties in the subset). This is particularly interesting, as when P increases, so does the average number of properties per entity, and hence achieving a high Micro-Recall is more challenging. Wiki2Prop achieves a MRR very close to 1 for $P \geq 15$. This means that for entities with at least 15 properties, the property predicted with the highest confidence was almost always correct.

Finally, Recoin has a better Micro-Precision than Wiki2Prop for $P \geq 20$, while having a very low Micro-Recall. This is a direct result of the Recoin algorithm: as it predicts the properties that are the most common in a class, entities that belongs to that class and have a large number of properties are very likely to have these *frequent* properties. Notably, we observe that the discriminating power of all ranking methods is high (instance-AUC > 0.95), where Wiki2Prop scores near perfect instance based ranking of the properties, achieving a new state of the art result on this problem.

4.5 Deployment

The resulting model of Wiki2Prop is deployed on ToolForge[11] for easier integration and querying. In contrast to the model trained for our evaluation, our deployed model is trained with the complete available dataset. It consists in an API service, which provides predictions for all entities present in the model. The service can be queried interactively [12]. The service also features a filter for already present entities based on the live version of Wikidata. The second component, depicted in Figure 1, is a so called *gadget*[13], which can be integrated with the Wikidata website itself. With the help of the gadget, Wikidata contributors can identify and complete missing properties.

4.6 Examples

To illustrate how Wiki2Prop outperforms statistical approaches such as Recoin, we look at some hand-picked examples originating from our predictions.

- **Johnny Depp**[14] has an actively maintained page in Wikipedia, and had 32 properties in Wikidata. Wiki2Prop recommends two additional properties *official website* and *sibling*. The former could be inferred from the class *Actor*, but the property *sibling* is clearly an entity specific information which can not be predicted through class based statistical analysis.
- **Star Trek: The Original Series**[15] has 36 properties in Wikidata; Wiki2Prop predicts 10 additional properties for it: *director*, *screenwriter*, *follows*, *original network*, *publication date*, *narrative location*, *main subject*, *executive producer*, *list of characters*, *aspect ratio*. From the proposed predicates, we observe that all except *follows* are relevant properties for the entity.
- **Ohio LinuxFest**[16] Is an entity that did not have an associate class at the time of our data collection. Wiki2Prop predicts *instance of*, *official website*, *country*, *logo image* and *coordinate location*, which are all sensible propositions.
- **Queenie (the waterskiing elephant)** This entity has (as of today[17]) a comparatively short, English only, Wikipedia article. Nevertheless, Wiki2Prop is able to identify the following relevant properties: *country of origin*, *image* and further still provides specific properties from the TV show domain as mentioned in the article: *original language of film or TV show*, *genre* and *publication date*.

A qualitative study on the effects originating from the added modality of images shows that diverse image features have a direct influence on specific Wikidata properties. The following selected examples illustrate potentially learned features from images. A strong effect can be seen on protein drawings boosting the properties in the same context. Further could we discover a correlation between black and white pictures of people and the prediction of the property *date of death*. Finally we observed that pictures of buildings and places significantly boost the property *coordinate location*.

5 CONCLUSION AND FUTURE WORK

In this work, we focused on the task of property prediction in a knowledge graph by leveraging textual and image information about the entities whenever available. We apply our work to Wikidata with the explicit goal to guide the community efforts in filling in missing information.

We tackle this problem by training neural networks to transfer information from pre-existing entity embeddings onto the space of predefined properties defined by Wikidata. To our knowledge, this is the first attempt to infer relevant properties from embeddings obtained from unstructured, multimodal and multilingual content. Our model, Wiki2Prop, improves upon existing type-based methods by modeling complex semantic information that is not necessarily captured by the type hierarchy alone. Moreover, and since the Wikidata structure is language-agnostic, we seamlessly combine multiple language and image resources to enrich entities regardless of their origin. Our experiments show that Wiki2Prop consistently outperforms state-of-the-art methods on a variety of metrics.

Our proposed method does not control for biases which are learned from the input data. Properties underrepresented in Wikidata, for example of a minority group, will have an effect on the prediction. We advice to control for potential biases dependent on the specific task solved with the predictions. In our presented use case of finding missing properties, bias effects are mitigated since human editors make the final decision on which information shall be added to Wikidata or not.

For future work, we plan to consider alternative sources of information, beyond Wikipedia and Wikimedia Commons. For example, the graph structure of Wikidata itself – whenever the entity is already *typed* and/or is connected to other entities in the graph – can be used, in addition to the fusion of further forms of external information such as audio and video, which could bootstrap or enrich the initial text-based embeddings we leverage. We also plan on deploying a property recommendation tool, based on Wiki2Prop predictions, that could directly be integrated into the Wikidata interface upon activation by any user.

Another promising avenue of future work would be to use our model in combination with information extraction methods; that is, our extended model could also generate candidate values in addition to predicting which entities and properties to focus on.

6 ACKNOWLEDGMENT

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 683253/GraphInt).

REFERENCES

[1] Ziawasch Abedjan and Felix Naumann. 2013. Improving RDF Data Through Association Rule Mining. *Datenbank-Spektrum* 13, 2 (01 Jul 2013), 111–120. https://doi.org/10.1007/s13222-013-0126-x

[2] Vevake Balaraman, Simon Razniewski, and Werner Nutt. 2018. Recoin: relative completeness in Wikidata. In *Companion Proceedings of the The Web Conference 2018*. 1787–1792.

[3] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* 41, 2 (2018), 423–443.

[4] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*

[11] https://toolforge.org/

[12] https://tools.wmflabs.org/wiki2prop/

[13] https://www.wikidata.org/wiki/Wikidata:Wiki2Prop

[14] https://www.wikidata.org/wiki/Q37175

[15] https://www.wikidata.org/wiki/Q1077

[16] https://www.wikidata.org/wiki/Q4043000

[17] https://en.wikipedia.org/wiki/Special:PermanentLink/996473264

(LREC'16). European Language Resources Association (ELRA), Portorož, Slovenia, 3967–3972.

[5] Serhat Selcuk Bucak, Rong Jin, and Anil K Jain. 2011. Multi-label learning with incomplete class assignments. In *CVPR 2011*. IEEE, 2801–2808.

[6] Yusuf Celik, Muhammed Talo, Ozal Yildirim, Murat Karabatak, and U Rajendra Acharya. 2020. Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images. *Pattern Recognition Letters* (2020).

[7] Guillem Collell, Teddy Zhang, and Marie-Francine Moens. 2017. Learning to predict: A fast re-constructive method to generate multimodal embeddings. *arXiv preprint arXiv:1703.08737* (2017).

[8] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 8609–8613.

[9] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2013. Large-scale linked data integration using probabilistic reasoning and crowdsourcing. *The VLDB Journal* 22, 5 (2013), 665–687.

[10] André Elisseeff and Jason Weston. 2002. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*. 681–687.

[11] Mariam Farda-Sarbas and Claudia Mueller-Birn. 2019. Wikidata from a Research Perspective – A Systematic Mapping Study of Wikidata. arXiv:1908.11153 [cs.DL]

[12] Johannes Frey, Marvin Hofer, Daniel Obraczka, Jens Lehmann, and Sebastian Hellmann. 2019. DBpedia FlexiFusion the Best of Wikipedia> Wikidata> Your Data. In *International Semantic Web Conference*. Springer, 96–112.

[13] Ignazio Gallo, Alessandro Calefati, and Shah Nawaz. 2017. Multimodal classification fusion in real-world scenarios. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 5. IEEE, 36–41.

[14] Lars C Gleim, Rafael Schimassek, Dominik Hüser, Maximilian Peters, Christoph Krämer, Michael Cochez, and Stefan Decker. 2020. SchemaTree: Maximum-Likelihood Property Recommendation for Wikidata. In *European Semantic Web Conference*. Springer, 179–195.

[15] Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Multi-modal models for concrete and abstract concept meaning. *Transactions of the Association for Computational Linguistics* 2 (2014), 285–296.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[17] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

[18] Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S Rosenblum. 2019. MMKG: multi-modal knowledge graphs. In *European Semantic Web Conference*. Springer, 459–474.

[19] Michael Luggen, Djellel Difallah, Cristina Sarasua, Gianluca Demartini, and Philippe Cudré-Mauroux. 2019. Non-parametric Class Completeness Estimators for Collaborative Knowledge Graphs—The Case of Wikidata. In *International Semantic Web Conference*. Springer, 453–469.

[20] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[22] Changsung Moon, Paul Jones, and Nagiza F Samatova. 2017. Learning entity type Embeddings for knowledge graph completion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2215–2218.

[23] Jose M Moyano, Eva L Gibaja, Krzysztof J Cios, and Sebastián Ventura. 2018. Review of ensembles of multi-label classifiers: models, experimental study and prospects. *Information Fusion* 44 (2018), 33–45.

[24] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 437–452.

[25] Y Nesterov. [n.d.]. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Math. Dokl*, Vol. 27.

[26] Daniel Oñoro-Rubio, Mathias Niepert, Alberto García-Durán, Roberto Gonzalez-Sanchez, and R. López-Sastre. 2019. Answering Visual-Relational Queries in Web-Extracted Knowledge Graphs. In *AKBC*.

[27] Natalia Ostapuk, D. Difallah, and P. Cudré-Mauroux. 2020. SectionLinks: Mapping Orphan Wikidata Entities onto Wikipedia Sections. In *Wikidata@ISWC*.

[28] Malte Ostendorff, Terry Ruas, Moritz Schubotz, Georg Rehm, and Bela Gipp. 2020. Pairwise Multi-Class Document Classification for Semantic Relations between Wikipedia Articles. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (Virtual Event, China) *(JCDL '20)*. Association for Computing Machinery, New York, NY, USA, 127–136. https://doi.org/10.1145/3383583.3398525

[29] Alessandro Piscopo and Elena Simperl. 2019. What We Talk About when We Talk About Wikidata Quality: A Literature Survey. In *Proceedings of the 15th International Symposium on Open Collaboration* (Skvde, Sweden) *(OpenSym '19)*. ACM, New York, NY, USA, Article 17, 11 pages. https://doi.org/10.1145/3306446.3340822

[30] Geoff Pleiss, Danlu Chen, Gao Huang, Tongcheng Li, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Memory-efficient implementation of densenets. *arXiv preprint arXiv:1707.06990* (2017).

[31] Radityo Eko Prasojo, Fariz Darari, Simon Razniewski, and Werner Nutt. 2016. Managing and Consuming Completeness Information for Wikidata Using COOL-WD. In *COLD@ISWC*.

[32] Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer, 55–69.

[33] Waseem Rawat and Zenghui Wang. 2017. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* 29, 9 (2017), 2352–2449.

[34] Simon Razniewski, Vevake Balaraman, and Werner Nutt. 2017. Doctoral advisor or medical condition: Towards entity-specific rankings of knowledge base properties. In *International Conference on Advanced Data Mining and Applications*. Springer, 526–540.

[35] L. Rettig, J. Audiffren, and P. Cudre-Mauroux. 2019. Fusing Vector Space Models for Domain-Specific Applications. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE Computer Society, Los Alamitos, CA, USA, 1110–1117. https://doi.org/10.1109/ICTAI.2019.00155

[36] Robert E Schapire and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine learning* 39, 2-3 (2000), 135–168.

[37] Jean Stawiaski. 2018. A pretrained densenet encoder for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*. Springer, 105–115.

[38] Yichuan Tang, Nitish Srivastava, and Ruslan R Salakhutdinov. 2014. Learning generative models with visual attention. In *Advances in Neural Information Processing Systems*. 1808–1816.

[39] Martin Wöllmer, Angeliki Metallinou, Florian Eyben, Björn Schuller, and Shrikanth Narayanan. 2010. Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In *Proc. INTERSPEECH 2010, Makuhari, Japan*. 2362–2365.

[40] Fei Wu, Raphael Hoffmann, and Daniel S. Weld. 2008. Information Extraction from Wikipedia: Moving Down the Long Tail. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. ACM, New York, NY, USA, 731–739. https://doi.org/10.1145/1401890.1401978

[41] Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2017. Image-embodied knowledge representation learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3140–3146.

[42] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 23–30.

[43] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 250–259.

[44] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[45] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250* (2017).

[46] Eva Zangerle, Wolfgang Gassler, Martin Pichl, Stefan Steinhauser, and Günther Specht. 2016. An Empirical Evaluation of Property Recommender Systems for Wikidata and Collaborative Knowledge Bases. In *Proceedings of the 12th International Symposium on Open Collaboration* (Berlin, Germany) *(OpenSym '16)*. ACM, New York, NY, USA, Article 18, 8 pages. https://doi.org/10.1145/2957792.2957804

[47] Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering* 18, 10 (2006), 1338–1351.

[48] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40, 7 (2007), 2038–2048.

[49] Min-Ling Zhang and Zhi-Hua Zhou. 2013. A review on multi-label learning algorithms. *TKDE* 26, 8 (2013), 1819–1837.

[50] Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. 2018. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, 100–107.