Learning to Map Wikidata Entities To Predefined Topics

Preeti Bhargava*
Demandbase
San Francisco, CA
pretsbhargava@gmail.com

Nemanja Spasojevic
Youtube
San Bruno, CA
sofra@alum.mit.edu

Sarah Ellinger
Juul Labs
San Francisco, CA
sarah.ellinger@lithium.com

Adithya Rao
Magic Leap
Sunnyvale, CA
adithyar@alumni.stanford.edu

Abhinand Menon
Beeswax
New York, NY
abhinandmenon@gmail.com

Saul Fuhrmann
Lime Bikes
San Francisco, CA
frmsaul@gmail.com

Guoning Hu
Amazon
Sunnyvale, CA
guoning.hu@gmail.com

ABSTRACT

Recently much progress has been made in entity disambiguation and linking systems (EDL). Given a piece of text, EDL links words and phrases to entities in a knowledge base, where each entity defines a specific concept. Although extracted entities are informative, they are often too specific to be used directly by many applications. These applications usually require text content to be represented with a smaller set of predefined concepts or topics, belonging to a topical taxonomy, that matches their exact needs. In this study, we aim to build a system that maps Wikidata entities to such predefined topics. We explore a wide range of methods that map entities to topics, including GloVe similarity, Wikidata predicates, Wikipedia entity definitions, and entity-topic co-occurrences. These methods often predict entity-topic mappings that are reliable, i.e., have high precision, but tend to miss most of the mappings, i.e., have low recall. Therefore, we propose an ensemble system that effectively combines individual methods and yields much better performance, comparable with human annotators.

KEYWORDS

entity topic mapping; entity topic assignment; natural language processing; knowledge base; wikipedia; wikidata

ACM Reference Format:
Preeti Bhargava, Nemanja Spasojevic, Sarah Ellinger, Adithya Rao, Abhinand Menon, Saul Fuhrmann, and Guoning Hu. 2019. Learning to Map Wikidata Entities To Predefined Topics. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3308560.3316749

*This work was done when all the authors were employees of Lithium Technologies | Klout

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316749

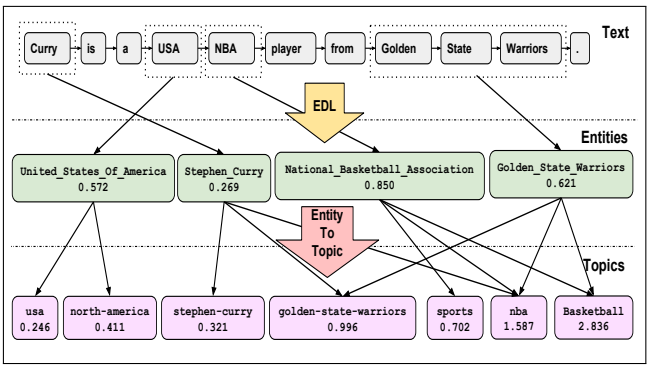


Figure 1: Topic extraction using Entity Disambiguation and Linking (EDL) together with entity-to-topic mapping.

1 INTRODUCTION

There have been many efforts to extract the rich information available in various types of user-generated text, such as webpages, blog posts, and tweets and represent it as a set of concepts which can then be used by various applications, such as content search, personalization, and user profile modeling. This can be achieved by understanding the *topics* in which users are interested or are experts [8, 16, 21, 22] by categorizing the user-generated text into a finite set of topics or categories.

Traditionally, statistical topic models such as LDA [7] have been used for topical categorization of text. These models are based on the idea that individual documents are made up of one or more topics, where each topic is a distribution over words. There have been many applications showing the power of these models on a variety of text documents (e.g. Enron emails, CiteSeer abstracts, Web pages). While LDA is a powerful tool for finding topic clusters within a document, it may miss implicit topics that are better suitable for document categorization.

Recently, tremendous advances have been made in entity disambiguation and linking (EDL) [1, 4, 9, 13, 14, 19]. Many EDL API services are now available to the public, including Google NLP[1],

[1]https://cloud.google.com/natural-language/

Watson Natural Language Understanding API[2], and Rosette Text Analytics[3]. These advanced EDL technologies and services make it practically elementary to extract a set of entities from a piece of text.

Unlike LDA-generated topics, entities are well defined concepts described in a knowledge base (KB) e.g. entities in the Wikidata KB. Modern KBs contain hundreds of thousands of entities or more. Some entities are quite broad, but more often they are very specific. When such narrow entities are extracted from text, they are somewhat informative, but they may be too specific and too many for the needs of a given application. Moreover, entities help enable a syntactic rather than a semantic understanding of the text. For example, in a search application where the query is "Golden State Warriors", documents indexed by the entity "Stephen Curry" are highly relevant, but may not be returned.

To address these challenges and meet the needs of general applications, a *topical taxonomy*, or hierarchical structure of topics, can be introduced. The primary advantage of using such a taxonomy rather than directly applying entities is to support product and business requirements such as:

(1) Limiting topics to a given knowledge domain.
(2) Imposing an editorial style or controlling the language used in describing topics (e.g., by imposing a character limit on topic names).
(3) Limiting the topic set in size so that an application user can better interact with the available topics. Topic taxonomy cardinality is orders of magnitude smaller than number of entities within the KB.
(4) Preventing unsuitable concepts from being represented as topics. These may include concepts that are:
   (a) Offensive or controversial (e.g. *Pornography*).
   (b) Either too general (e.g. *Life*) or too specific (e.g. *Australian Desert Raisin*).
   (c) Redundant with one another (e.g. *Obamacare* and *Affordable Care Act*).

For example, Klout.com[4] used a custom taxonomy[10] which was modeled around capturing topics of social media content in order to build topical user profiles [21, 22]. Another example is Google Adwords, which uses a small, human-readable taxonomy[5] to allow advertisers to target personalized ads to users based on topical interests.

Thus, to categorize text into topics, one can take advantage of mature EDL systems by *mapping* the entities extracted from the text to topics in a topical taxonomy. An example of using EDL to extract topics is shown in Figure 1. Although there have been studies that touch upon aspects of the entity-topic mapping problem, either while modeling the relationships among entities or while modeling the concepts of entities, no systematic study exists of this particular task. However, we find that an ensemble of some selected models is able to yield very good results.

Our main contributions in this paper are:

- We propose a system that maps entities in a KB (derived from Wikidata) to topics in a taxonomy. Together with EDL, our system allows one to extract the concepts that best meet specific application needs from a given text.
- We study multiple popular models that explore the relationship among entities from various perspectives, including cooccurrence, word embeddings, and Wikipedia content. We find that each of them performs reasonably well on mapping entities to topics.
- We investigate multiple approaches to combine the above models into a stacked ensemble model and obtain much better results. We find that best performance is achieved through a SVM meta-model (AUC: 0.874 and F1: 0.786) which yields results comparable to human annotators.
- We show that although our system is developed with a specific topical taxonomy, one can easily adapt it for use with other taxonomies.
- Open data - we make our label set publicly available.

2 PROBLEM SETTING

In this work, we attempt to build a system that maps entities in an entity set to topics in a topic set.

2.1 Entity set

Wikidata is the largest free and open KB, acting as a central structured data store of all Wikimedia content. Entities are the atomic building blocks of Wikidata. Information about a Wikidata entity is organized using named predicates, many of which annotate relations between entities.

We derived our entity set from Wikidata for the following reasons:

- Wikidata entities are widely used in the community, allowing our system to benefit a large audience.
- Wikidata contains more than $43M$ entities, covering the majority of concepts that people care about. These entities include people, places, locations, organizations, etc. There are entities for broad concepts, such as *Sports*, and entities for very specific concepts, such as *5th Album of The Beatles*.
- Wikidata entities come with rich annotations that can be utilized for this problem. For example:
  — Every entity is linked to a corresponding word or phrase in multiple languages.
  — Millions of predicates describe special relations among entities (see Section 5.2.1 for more details).
- There are datasets associated with Wikidata that provide useful information. In this work, we leverage Wikipedia pages (see Section 5.3.1 for more details) and DAWT [20], an extension dataset to Wikidata.

2.2 Topic set

Topics that entities are mapped to are application specific. In this study, we use a topic set from the Klout Topic Ontology (KTO) [10] which itself is a subset of Wikidata. KTO contains about 8K topics. Each topic is annotated with two types of annotations:

- **Wikidata id** - Since KTO is a subset of Wikidata, each topic is equivalent to a Wikidata entity which is referred to as the

[2]https://www.ibm.com/watson/services/natural-language-understanding/
[3]https://developer.rosette.com
[4]Klout.com has now been shut down since May 25, 2018
[5]https://support.google.com/ads/answer/2842480

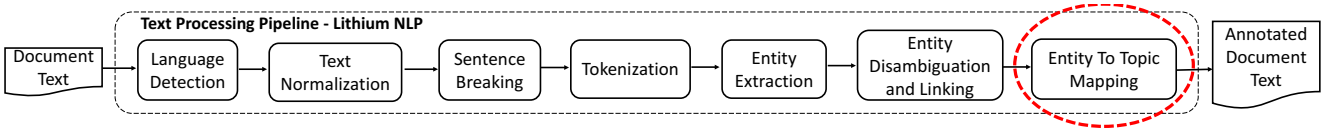


Figure 2: The Lithium NLP pipeline.

primary entity for that topic. The wikidata id for the *primary entity* is used as this annotation.

- **Parent and child topics** - Topics are organized hierarchically into multiple levels. Topics at a higher level are broader and usually link to multiple narrower topics at the next lower level. For example, topics *Entertainment*, *Pop music* and *The Beatles* have the following relationship within KTO:

  Entertainment ⇒ *Pop music* ⇒ *The Beatles.*
  This hierarchical relationship is encoded in these parent and child topic annotations.

To define our task formally, let $\mathcal{E}$ be Wikidata entity set. Let $\mathcal{T}$ be a topic set with a hierarchical structure. Let $e \in \mathcal{E}$ be an entity and $t \in \mathcal{T}$ be a topic. For any t, there is an equivalent entry in $\mathcal{E}$, which is referred to as the primary entity, e_t. Note that an entity, whether or not it is a primary entity, can map to multiple topics. To be concise, if e shall be mapped to t, we say t is relevant to e; otherwise, we say that t is irrelevant to e. Thus, the problem we attempt to solve is:

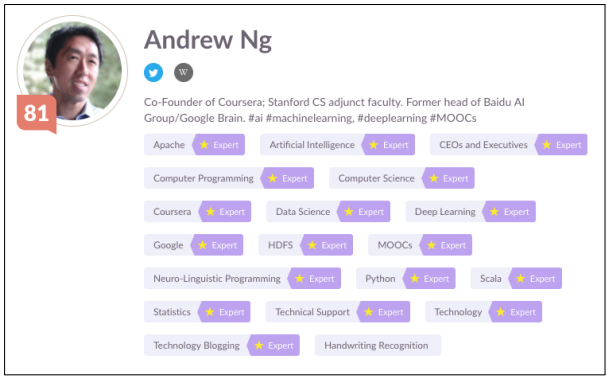
Given a pair of (e, t), determine whether or not t is relevant to e irrespective of whether t is equivalent to e

This is a binary classification problem. For practical purposes, we aim to build a regression system that not only performs classification, but also yields a quantitative measure of how relevant e is to t, which is often useful for a subsequent application stage.

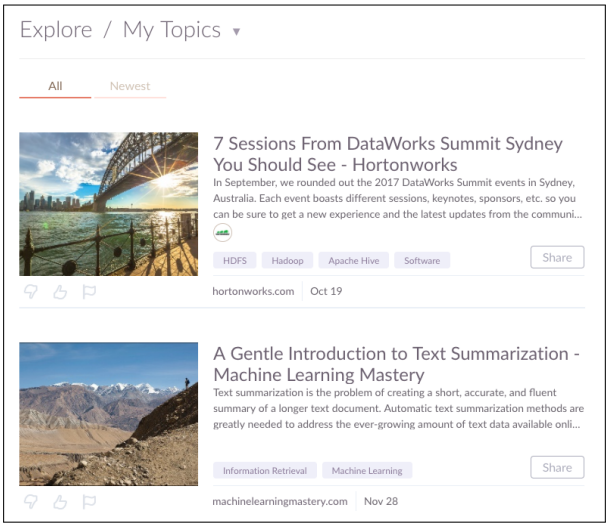
3 APPLICATIONS

The described system is a part of the Lithium NLP [6] [5] pipeline - a resource-constrained, high-throughput and language-agnostic system for information extraction from noisy user generated text such as that available on social media. Figure 2 shows a partial view of the Lithum NLP pipeline, where the sub-module 'Entity To Topic Mapping' is the final stage. Lithium NLP is capable of extracting a rich set of information including entities, topics, hashtags and sentiment. Lithium NLP currently supports multiple languages including Arabic, English, French, German, Italian and Spanish. It supports large scale data from several social media platforms such as Twitter, Facebook, Linkedin, etc. by processing about $500M$ new social media messages, and $0.5M$ socially relevant URLs shared daily. Since it employs statistical NLP techniques, it uses the large scale of the data to help overcome the noisiness.

In the Lithium NLP pipeline, entity-to-topic mapping is used to convert a set of Wikidata entities to a set of topics that are most relevant to a piece of text. These topics are stored as annotations, which are consumed by multiple Lithium products for various tasks, such as indexing content and building user profiles. One notable application was Klout[7] - a consumer platform which integrated users' data from multiple social networks in order to measure their online social influence via the *Klout Score*[8] [18]. On Klout, these topic mappings were used to model users' topics of interest [22] and expertise [21] in order to recommend personalized content to the users. Figure 3a shows a user's topical profile on Klout, and Figure 3b shows content recommendations derived from those topics. In addition, these topical profiles were included in Twitter's PowerTrack APIs[9].



(a) Andrew Ng's inferred expertise topics.



(b) Andrew Ng's content feed, personalized to his topics.

Figure 3: Topic mapping enabled applications.

[6] http://nlp.app.lithium.com/ui

[7] https://klout.com

[8] https://klout.com/corp/score

[9] http://support.gnip.com/enrichments/klout.html

Lithium NLP currently enables text and user annotations within Lithium's social media management tools[10], and is used to analyze $20 + M$ new daily engagements across Lithium's 400+ communities[11].

4 DATA SETS

As described in Section 2, we use an entity set derived from Wikidata as our entity set and KTO as our topic set. Wikidata entities were limited to the $1M$ most important ones, where importance measure is described in Bhattacharyya and Spasojevic [6]. To train our ensemble, we collected a dataset of $26.6K$ entity-topic pairs labeled as 'Relevant' if the topic is relevant to the entity or 'Irrelevant' otherwise. We had 3 data sources - a control dataset which we used to control quality of operator evaluations and two labeled datasets that we used for training, validation, and testing.

4.1 Control Set

To control the quality of labeling done by annotators, a small set of 100 labeled $(entity, topic)$ pairs was generated by an in-house expert panel. 10% of each batch of tasks prepared for human annotators included data taken from this control set; based on how accurate that human's evaluations were for control-set pairs, we were able to estimate the quality of the batch as a whole.

4.2 Amazon Mechanical Turk

We collected about $8.6K$ annotations via Amazon Mechanical Turk (AMT). Workers were shown an entity and its description as well as a topic and its description; they were then asked to identify whether the entity hierarchically mapped to the topic, or was unrelated. The entities were picked from the top $100K$ entities of our KB. For each entity-topic pair within Wikidata and its predicate, we sampled pairs so that predicate distribution was balanced. This guaranteed that a diverse set of relationships was present in our data set.

Each AMT task had about 100 such entity-topic pairs and was given to one unique worker, who was compensated $3. Before doing large-scale data collection, we calibrated the workers on accuracy and time via a pilot set using 3 workers. The workers exhibited an accuracy of 80% when compared to the control set labels. Hierarchical relationships between entities and topics is a subjective measure and as such there may not be total agreement among the workers. Hence, we also computed the consensus of workers on the pilot test data and found that they showed a consensus of 81%.

4.3 In-House Labeling

We also collected labeled data from 5 team members labeling about $18K$ entity topic pairs. In this case we sampled data from the top $1M$ entities where the probability of selecting an entity was inversely proportional to its rank. This ensured that we have higher representation of top-ranked entities, which are more important to our application. As with Mechanical Turk, we did a pilot test to see how accurately team members labeled the data. Our in-house experts achieved an accuracy of 89% on the pilot test and a consensus of 82%.

[10] https://www.lithium.com/products/social-media-management/
[11] https://www.lithium.com/products/online-communities/

5 METHODOLOGY

Figure 4 shows a high level overview of our entity-topic mapping pipeline. The pipeline takes an (entity, topic) pair and yields a measure of how relevant the topic is to the entity. To build this system, we first considered a wide range of models that approached this problem from different aspects. Some of them generated statistical metrics on mapping an entity to a topic. Others were models that had been applied to similar problems erstwhile.

We then combined these models using a stacked ensemble as this helps overcome the inherent biases and lack of coverage (manifested as low recall) of the individual models. To make the system run efficiently in production, we removed models that were either too computationally intensive or had insignificant contributions to the performance[12]. The final ensemble combines 8 models, which are described below.

As a running example through this section, we will use the entity *Q1414593 - Patient Protection and Affordable Care Act*. Table 1 shows the mapping scores generated by the 8 models between this entity and topics that are most relevant to it.

5.1 Word Embedding Based Methods

5.1.1 GloVe Model. Pennington et al. [15] introduced Global vectors for word representation (GloVe) - a technique that generates word embeddings, i.e., vectors of real numbers, for representing the linguistic contexts of words. These embeddings allow the derivation of quantitative distance metrics, such as cosine similarity, to describe semantic "distance" between words.

As part of this model, we calculate the cosine similarity of embeddings for each (entity, topic) pair by using the primary entity for the topic. Thus, given a topic t, let e_t be the corresponding primary entity, we then have a score S measuring the similarity between t and an entity e as

$$S(t, e) = S(e_t, e). \tag{1}$$

5.1.2 GloVe Parents model. Although an entity and a relevant topic tend to have high word embedding based similarity, an entity and an irrelevant topic may often have high similarity too. Take the example of the entity *San Francisco* and the topics *Los Angeles* and *California*. Similarity between *San Francisco* and *Los Angeles* is 0.66, whereas that between *San Francisco* and *California* is 0.55. While this similarity metric indicates *Los Angeles* as a better topic for *San Francisco*, because both *Los Angeles* and *San Francisco* are cities in *California*, their relationship is not hierarchical, which is why *California* would be the only acceptable topic.

Such hierarchical relationships are not well represented in GloVe vectors, which rather capture peer to peer similarity. However, these hierarchical relationships are implicitly represented in our topic set $\mathcal{T}$. In particular, our topic set is structured as a directed acyclic graph, where the edges between topics represent hierarchical relationships. Topics in a higher level are broader and usually link to multiple narrower child topics in the next lower level. For example, both *Los Angeles* and *San Francisco* are child topics of *California*.

Therefore, we combine embeddings with the hierarchy of our topic set $\mathcal{T}$ to build a new measure. We first pick the most relevant

[12] Due to lack of space, we are unable to describe the models that we removed.

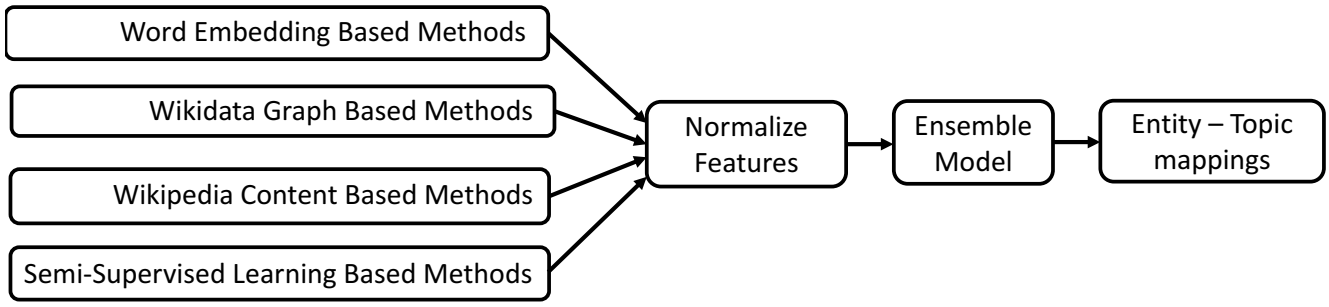


Figure 4: High level overview of the entity-topic mapping pipeline.

Table 1: Entity-topic mappings generated for the entity "Patient Protection and Affordable Care Act"

Model	Health Care	Health Insurance	U.S. Presidents	Politics	Affordable Care Act
GloVe	0	0	0.462	0.535	1.0
GloVe Parents	0.264	0.755	0.277	0.298	0
Wikidata Hierarchical Predicates	0	0	0	0	1.0
Wikidata Hierarchical Location Predicates	0	0	0	0	0
Wiki Pages Content	0.866	0	0.701	0.338	1.0
Wiki Pages Content Parents	0.132	0.868	0.229	0.736	0
Frequency Adjusted Co-occurrence	0	0	0	0.447	1.0
Topic Normalized Co-occurrence	0	0	0	0	1.0
Combined Ensemble (SVM)	0.905	0.904	0.896	0.896	1.0

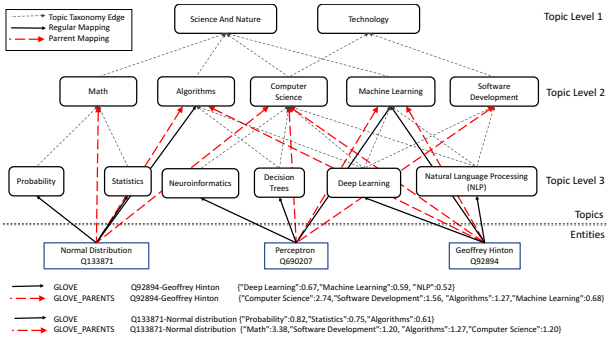


Figure 5: Higher-level topics inferred using the topic hierarchy.

topics via above similarities. Let V be the set of K topics that is most relevant to entity e. That is,

$$S(t_i, e) \geq S(t_j, e) \; \forall \; t_i \in V, t_j \notin V \tag{2}$$

We then calculate a combined similarity score of a parent topic t_p from the similarity scores of its child topics. Let C be the set containing all child topics of t_p. The combined similarity score between t_p and e is:

$$S(t_p, e) = \sum_{t_i \in C \cap V} S(t_i, e) \tag{3}$$

Parameter K is obtained through cross-validation and best performance is obtained when $K = 10$.

As an example, Figure 5 demonstrates how we leverage the topic hierarchy to infer the higher level topics for an entity. For entity *Q133871-Normal distribution*, the GloVe word vector model generates a ranked list of topics as: { *Probability* : 0.82, *Statistics* : 0.75, *Algorithms* : 0.61 .... }. Topic *Probability* and *Statistics* have parent topic *Math*. In the GloVe word vectors parents model, we aggregate the strengths from the children topics to infer {*Math* : 3.38, *Software Development* : 1.20, *Algorithms* : 1.27, *Computer Science* : 1.20....} for the entity *Q133871-Normal distribution*.

5.2 Wikidata Graph Based Methods

5.2.1 Wikidata Hierarchical Predicates Model. In this model, we leverage the structure of the Wikidata graph to infer hierarchical relationships between entities and topics. Although there are thousands of predicates defined in Wikidata, most (entity, topic) pairs have zero or one predicate. As a result, we only observe 146 unique predicates between entities and topics in our data set. In addition, we notice that location-related topics are overrepresented in this set. Hence, we consider only non-loccation predicates.

First, for each entity topic pair (e, t), let us define a predicate vector as $P(e, t) = \{r_i(e, t)\}, i \in [1, 146]$. $r_i(e, t)$ is 1 if the corresponding predicate connects e and t and 0 otherwise. We then apply a logistic regression model to estimate the probability of mapping e to t from predicate vector:

$$H(e, t) = \frac{1}{1 + e^{-\theta^t P(e,t)+b}} \tag{4}$$

Weight vector θ and bias b are estimated using the training dataset discussed in Section 8. As expected, individual weights assigned to predicates are fairly interpretable such that predicates with high score tend to correspond to hierarchical relations, such as "Instance Of" (P17), "Subclass of" (P279) etc.

5.2.2 WikiData Hierarchical Location Predicates Model. In this model we consider only the location predicates which are excluded from the previous model. This new location-based model is a Logistic Regression model.

5.3 Wikipedia Content Based Methods

5.3.1 Wikipedia Pages Content Model. The Wikipedia page of an entity usually gives a definition that contains related topics in the first paragraph. Therefore, we derive a measure that links an entity to topics based on its Wikipedia definitions in multiple languages.

Given an entity e, we first extract primary entities from the first paragraph of the corresponding Wikipedia page P_l for a given language l, via the EDL algorithm described in Bhargava et al. [4]. As an example, below is the first paragraph of the entity *machine_learning* in the English Wikipedia with extracted primary entities in bold:

*Machine learning is a field of **computer science** that gives **computers** the ability to learn without being explicitly programmed.*

Thus, for a topic t with primary entity e_t, the similarity score (per language) is defined as

$$S_l(t,e) = \begin{cases} \frac{1}{d(e_t)} & if\ e_t \in P_l \\ 0 & else \end{cases} \tag{5}$$

where $d(e_t)$ is the distance from the n-gram representing e_t to the beginning of the paragraph. The derived measure is an aggregation of the above score across all languages:

$$S(t,e) = \sum_l S_l(t,e) \tag{6}$$

Leveraging this across multiple languages helps boost most relevant topics for a given page. Note that since Wikipedia pages often contain location topics (eg. place of birth, location of headquarters, etc), which may often be irrelevant, we remove all the location topics in this measure.

5.3.2 Wikipedia Pages Content Parents Model. Similar to the GloVe word vector parents model described in Section 5.1.2, we combine the output of the Wikipedia pages content model with the inherent hierarchy of our topic set $\mathcal{T}$. Given an entity e, let V_d be the set of K topics that is most relevant to e. That is,

$$S(t_i,e) \geq S(t_j,e)\ \forall\ t_i \in V_d, t_j \notin V_d \tag{7}$$

We calculate a combined score of a parent topic t_p from the wiki-content scores of its child topics. Again, let C be the child topic set of topic t_p. The combined similarity score between t_p and e is:

$$S(t_p,e) = \sum_{t_i \in C \cap V} S(t_i,e) \tag{8}$$

5.4 Semi-Supervised Learning Based Methods

Our final model for entity to topic mapping utilizes co-occurrence frequencies for entities obtained from our DAWT [20] dataset. We capture co-occurrence frequencies among entities by counting all the entities that simultaneously appear within a sliding window of 50 tokens. Moreover, this data is accumulated across all languages and is language-independent in order to better capture relations and create a smaller memory footprint when supporting additional languages. Also, for each entity, we consider only the top 30 co-occurring entities which have at least 10 or more co-occurrences across all supported languages.

5.4.1 Frequency Adjusted Co-occurrence. Co-occurrence count has its own shortcomings as a measure of quantifying relationship between entities, because some entities occur more frequently in the dataset than others and hence will co-occur more with other entities. Therefore, we adjust the co-occurrence counts with entity frequencies. Let e denote an entity, $C(e)$ denote the occurrence count of e, N denote the total number of entities, and $C(e_i,e_j)$ denote the co-occurrences count of entity e_i and e_j. Then, the frequency adjusted co-occurrence of e_i and e_j i.e. $C_f(e_i,e_j)$ is calculated as:

$$C_f(e_i,e_j) = C(e_i,e_j) \cdot \log\left(\frac{N}{C(e_j)}\right) \tag{9}$$

Thus, given a topic t and e_t as its equivalent primary entity, we have a measure of linking an entity e to a topic t via $C_f(e,e_t)$.

5.4.2 Topic Normalized Co-occurrence. Some topics happen more frequently than other topics. As a result, frequency adjusted co-occurrences tend to favor topics that occur often in our corpus and knowledge base. To address this problem, we introduce a measure that further normalizes frequency adjusted co-occurrence for each topic. In particular, $C_t(t,e)$, normalized co-occurrence of topic t and entity e, is calculated as:

$$C_t(t,e) = \frac{C_f(t,e)}{\sum_i C_f(t,e_i)} \tag{10}$$

In this model, for each primary entity, we use the *co-occurring entity* with the strongest weight for the frequency adjusted co-occurrence. We then aggregate the topics for each *co-occurring entity* by mapping the primary entity to its equivalent topic. These give us the most relevant topics for the *co-occurring entity* with the normalized frequency adjusted co-occurrence score as the strength of the entity topic mapping.

5.5 Stacked Ensemble Model

To frame this as a machine learning problem, we define a feature as a numerical score associated with how strongly an entity should map to a topic in a hierarchical manner. Each of the models described above defines a separate feature. Our goal is to combine these features in a stacked ensemble to give the least error or loss in terms of entity to topic mappings.

We wish to compute a score for each entity-topic pair, trying to estimate the relevance of the topic to the entity. We define a feature vector $\mathcal{F}(t,e)$ for a entity e and a topic t as:

$$\mathcal{F}(t,e) = [f_1(t,e), f_2(t,e), ..., f_m(t,e)] \tag{11}$$

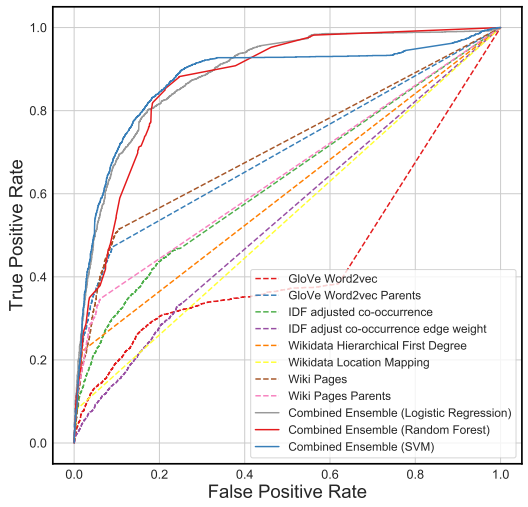


Figure 6: ROC curve on test set.

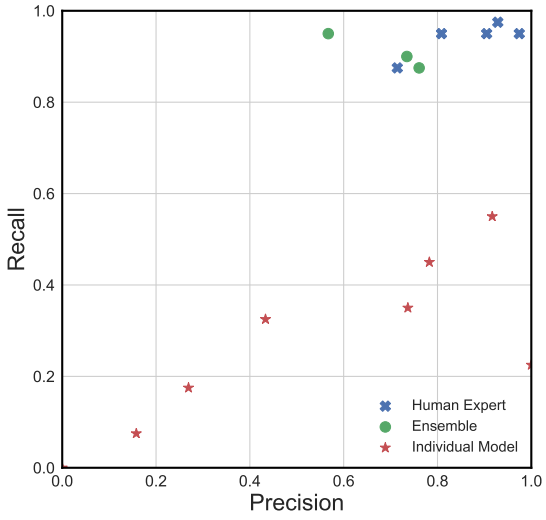


Figure 7: Precision and recall on control set.

where $f_k(t,e)$ is the feature value associated with a specific model which were introduced earlier. The normalized feature values are denoted by $\hat{f}_k(t,e)$ and the normalized feature vector is represented as: $\hat{\mathcal{F}}(t,e) = [\hat{f}_1(t,e), \hat{f}_2(t,e), ..., \hat{f_m}(t,e)]$. Normalization was calculated as:

$$\hat{f}_k(t,e) = \frac{f_k(t,e)}{\max_{t_j \in \mathcal{T}} f_k(t_j,e)} \tag{12}$$

All training has been performed on normalized feature vectors.

We then treat this task as a binary classification problem and apply the following methods: logistic regression, random forest, and support vector machine (SVM). The results are summarized in the Section 6.

6 EVALUATION AND RESULTS

In Table 2 we show the performance of individual models as well as ensemble models. The evaluations were performed on the reserved test set, which represented 20% of the labeled dataset not including the *control set*. We can see that each individual model performs reasonably well on this task. The best performing models, *Wikipedia Pages* and *GloVe Parents*, had an AUC of 0.721 and 0.701 respectively, while the worst performing model was *GloVe* with an AUC of 0.449. The poor performance of GloVe can be explained by the fact that GloVe embeddings capture similarity well, but not hierarchy.

At the bottom of Table 2 we show the performance of the ensemble models. All of the ensemble models had roughly similar performance, and significantly outperformed individual models. The SVM ensemble performed the best with a F1 of 0.786 and an AUC of 0.874. The ROC curves for individual and ensemble models are shown in Figure 6. We notice also that the majority of performance deterioration for individual models is caused by low recall. All of the models have recall less than 0.325, while the best ensemble achieved recall of 0.752. This means that we have chosen a diverse set of models, complementary to each other, which when combined result in high performance.

In Figure 7 we plot precision and recall for individual models, ensemble models, and expert human annotators evaluated on the *control* introduced in Section 4.1. We can see that the ensemble models were comparable to human annotator performance, while individual models were less successful.

7 LESSONS LEARNED

Some of the challenges we faced when implementing our entity to topic mapping system were:

- **Ambiguity of Relationships.** Topic mapping is very subjective; the perceived strength of the relationship between an entity and a topic can vary from human to human, depending on their domain knowledge and the intended application. For example, should the music group *Deep Purple* map to *Progressive Rock*, although their musical genre shifted over time toward *Heavy Metal*? These examples demonstrate the subjectivity and ambiguity of the problem.
  In addition, we found that Mechanical Turk annotators reached only 81% consensus, with in-house annotators reaching 89% when measured on the *control set*, thus, demonstrating the ambiguity.
- **Similarity vs. Hierarchy.** Many models successfully capture similarity (peer-to-peer relations) between an entity and a topic; however, parent-child relations are much harder to capture. To minimize false positives, we introduced the *Parent* models that take a base model and roll up candidate topics to their parents within the topic taxonomy.
- **Knowledge Base Biases.** Many KBs are rich with factual data; however, some entity subsets may be overrepresented for the purposes of an application. For example, we notice

Table 2: Performance metrics for the different models.

Model	F1	AUC	Recall	Precision	Accuracy
GloVe	0.333	0.449	0.232	0.590	0.606
GloVe Parents	0.429	0.701	0.282	0.895	0.681
Wikidata Hierarchical Predicates	0.286	0.599	0.170	0.917	0.641
Wikidata Hierarchical Location Predicates	0.142	0.536	0.077	0.912	0.605
Wiki Pages Content	0.456	0.721	0.312	0.852	0.685
Wiki Pages Content Parents	0.328	0.641	0.203	0.850	0.647
Frequency Adjusted Co-occurrence	0.440	0.635	0.325	0.680	0.649
Topic Normalized Co-occurrence	0.224	0.553	0.142	0.530	0.583
Combined Ensemble (Logistic Regression)	0.760	0.887	0.695	0.839	0.806
Combined Ensemble (Random Forest)	0.757	0.871	0.735	0.780	0.791
Combined Ensemble (SVM)	0.786	0.874	0.752	0.824	0.819

that in Wikipedia and Wikidata entities locations are frequently described in even non-geographic entities. Although still informative for our task, the frequency of these relations had a negative impact on final results. To address this problem we must either filter location-based topics or split a single model into two disjoint models, where one represents mappings purely to location-related topics and other represents mappings to rest of taxonomy.

- **Taxonomy Constraints.** In our problem statement we said that topic taxonomy is subset of KB entities, and based on this principle, we heavily rely on a 1:1 mapping of a topic to its primary entity. However, depending on the taxonomy used, there may be cases where a topic does not have a representative entity, in which case a workaround would have to be devised.

8 OPEN DATASET

The dataset used to run evaluations and build models has been opened at **https://github.com/klout/opendata**. It includes $26.6K$ triplets $(entity, topic, label)$ where for a given pair $(entity, topic)$ $label$ indicates if $entity$ should map to given $topic$ or not. A detailed explanation of how data has been sampled and how labels have been generated can be found in Section 4. Each entity is represented with a Wikidata[13] id and each topic with a Klout Topic Ontology id [10]. For convenience, the dataset also includes a display name.

9 RELATED WORK

Statistical topic modeling such as LDA [7] is based on the premise that individual documents are made up of one or more topics and each topic is a distribution over words. Newman et al. [12] adapt LDA to model associations between entities and topics that occur in a document. Kim et al. [11] propose a topic model for analyzing a collection of documents with given entities and model the correlation between entity-term and topic-term distributions.

However, we are primarily interested in finding mapping a set of entities to a set of topics regardless of the documents that the entities appear in. We can think of these mappings as more global mappings that are not local to the documents and do not depend on the document. This problem has not received significant attention in the industry. Balog [2] presents a model of entity-topic mapping in order to identify topical experts but they focus on only 2 types of entities - people and moods. Balog et al. [3] extend this work to identify entity topic mappings in online news articles. Raghavan et al. [17] explored entity-entity modeling and relationship description but mostly focused on entities that co-occur together in a text window. This is similar to the semi-supervised learning models in our ensemble model.

Our work differs from these previous as we attempt to build a generic system that maps Wikidata entities to any predefined topic taxonomy.

[13] https://www.wikidata.org

10 CONCLUSION AND FUTURE WORK

In this paper, we addressed the problem of mapping Wikidata entities to predefined topics. We built an ensemble model that leverages GloVe word vector similarity, Wikidata predicates, Wikipedia entity definitions, and entity-topic co-occurrences for mapping entities to topics. Our system obtains performance comparable to human annotators and has been integrated as part of the Lithium NLP pipeline, serving multiple applications running in production. We continually collect production data and customer feedback to further improve the system. In future, we plan on incorporating multiple KBs to handle Knowledge Base bias. In addition, as this research was done using a single topic set, we would like to test against other domain-specific taxonomies, and extend our system to handle cases where there is no 1:1 topic to entity mappings.

REFERENCES

[1] Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio. 2014. Automatic Creation of Arabic Named Entity Annotated Corpus Using Wikipedia. Association for Computational Linguistics.

[2] Krisztian Balog. 2007. People search in the enterprise. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 916–916.

[3] Krisztian Balog, Maarten de Rijke, Raymond Franz, Hendrike Peetz, Bart Brinkman, Ivan Johgi, and Max Hirschel. 2009. Sahara: Discovering entity-topic associations in online news. In *Proceedings of ISWC*, Vol. 9.

[4] Preeti Bhargava, Nemanja Spasojevic, and Guoning Hu. 2017. High-Throughput and Language-Agnostic Entity Disambiguation and Linking on User Generated Data. In *Proceedings of WWW 2017 workshop on Linked Data on the Web*.

[5] Preeti Bhargava, Nemanja Spasojevic, and Guoning Hu. 2017. Lithium NLP: A System for Rich Information Extraction from Noisy User Generated Text on Social Media. In *Proc. of the 3rd Workshop on Noisy User-generated Text*. 131–139.

[6] Prantik Bhattacharyya and Nemanja Spasojevic. 2017. Global Entity Ranking Across Multiple Languages. In *Companion Proceedings of the WWW*. 761 – 762.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[8] Christopher S Campbell, Paul P Maglio, Alex Cozzi, and Byron Dom. 2003. Expertise identification using email communications. In *Proc. of ACM Conference on Information and Knowledge Management (CIKM)*.

[9] Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data.. In *EMNLP-CoNLL*, Vol. 7. 708–716.

[10] Sarah Ellinger, Prantik Bhattacharyya, Preeti Bhargava, and Nemanja Spasojevic. 2017. Klout topics for modeling interests and expertise of users across social networks. *arXiv preprint arXiv:1710.09824* (2017).

[11] Hyungsul Kim, Yizhou Sun, Julia Hockenmaier, and Jiawei Han. 2012. Etm: Entity topic models for mining documents associated with entities. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 349–358.

[12] David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. 2006. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 680–686.

[13] Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. AIDA-light: High-Throughput Named-Entity Disambiguation.. In *LDOW'14*.

[14] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2012. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence* 194 (2012), 151–175. https://doi.org/10.1016/j.artint.2012.03.006

[15] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation.. In *Proceedings of EMNLP*. 1532 – 1543.

[16] Ana-Maria Popescu, Krishna Yeswanth Kamath, and James Caverlee. 2013. Mining Potential Domain Expertise in Pinterest.. In *UMAP Workshops*.

[17] Hema Raghavan, James Allan, and Andrew McCallum. 2004. An exploration of entity models, collective classification and relation description. In *KDD Workshop on Link Analysis and Group Detection*. 1–10.

[18] Adithya Rao, Nemanja Spasojevic, Zhisheng Li, and Trevor Dsouza. 2015. Klout score: Measuring influence across multiple social networks. In *IEEE Intl. Conf. on Big Data*.

[19] Delip Rao, Paul McNamee, and Mark Dredze. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*. Springer, 93–115.

[20] Nemanja Spasojevic, Preeti Bhargava, and Guoning Hu. 2017. DAWT: Densely Annotated Wikipedia Texts across multiple languages. In *Companion Proceedings of WWW*. 1655 – 1662.

[21] Nemanja Spasojevic, Prantik Bhattacharyya, and Adithya Rao. 2016. Mining half a billion topical experts across multiple social networks. *Social Network Analysis and Mining* 6, 1 (2016), 1–14.

[22] Nemanja Spasojevic, Jinyun Yan, Adithya Rao, and Prantik Bhattacharyya. 2014. LASTA: Large Scale Topic Assignment on Multiple Social Networks. In *Proc. of ACM KDD*. 1809 –1818.