Automatic Construction of Enterprise Knowledge Base

Junyi Chai, Yujie He, Homa Hashemi, Bing Li,
Daraksha Parveen, **Ranganath Kondapally**, **Wenjin Xu**
Microsoft Corporation
juchai,yujh,hohashem,libi,daparvee,rakondap,wenjinxu@microsoft

Abstract

In this paper, we present an automatic knowledge base construction system from large scale enterprise documents with minimal efforts of human intervention. In the design and deployment of such a knowledge mining system for enterprise, we faced several challenges including data distributional shift, performance evaluation, compliance requirements and other practical issues. We leveraged state-of-the-art deep learning models to extract information (named entities and definitions) at per document level, then further applied classical machine learning techniques to process global statistical information to improve the knowledge base. Experimental results are reported on actual enterprise documents. This system is currently serving as part of a Microsoft 365 service.

1 Introduction

Massive knowledge bases constructed from public web documents have been successful in enriching search engine results in Bing and Google for over a decade (Noy et al., 2019). There is growing interest in automatically constructing a similar knowledge base for each enterprise from their internal documents (e.g., web pages, reports, emails, presentation decks; textual contents in natural language form are all referred to as documents in this paper). Such knowledge base can help an enterprise to better organize its domain knowledge, help employees (users) better find and explore knowledge, and to encourage knowledge sharing.

Mining knowledge from enterprise documents poses unique challenges. One challenge is that the system needs to be fully automated without per enterprise customization or existing (semi-) structured sources. Knowledge base construction from web documents is often based on bootstrapping entities from human-curated sources such as Wikipedia with customized extraction rules (DBpedia: Auer et al., 2007, Freebase: Bollacker et al., 2008, YAGO2: Hoffart et al., 2013), or the existence of a prior knowledge base (Knowledge Vault: Dong et al., 2014). Maintaining such Wiki site and keep it fresh is costly for enterprise. Another challenge is that most training data for natural language processing (NLP) models is from public documents. Enterprise documents can have different writing style and vocabulary than the public documents. The data distributional shift (Quiñonero-Candela et al., 2008) is a challenge as (a) we need model to generalize better to enterprise domain and (b) we need test metrics to reflect the actual performance on enterprise documents to guide model development.

On the other hand, enterprise domain brings new opportunities. For search engines, the knowledge base must be extremely accurate. This requirement limits the usage of NLP models to extract information from unstructured text as few models can achieve the required precision with meaningful coverage. In enterprise domain, we can relax the requirement on accuracy as enterprise users are expected to spend more time to absorb and discriminate information. In addition, users can curate and improve the automatically constructed knowledge base, which is not an option for search engine users. The relaxation on accuracy requirement makes it possible to perform knowledge mining on unstructured text by heavily relying on NLP techniques.

In this paper, we present the first large-scale knowledge mining system for enterprise documents taking advantage of recent advances in NLP such as pretrained Transformers (Devlin et al., 2019) as well as traditional NLP techniques. It is in production since February 2021 as part of a Microsoft 365 feature (Microsoft Viva Topics[1]). For an enterprise that enables this feature, our system will build a knowledge base from its internal documents that already exist in Microsoft cloud and will keep it

[1] https://www.microsoft.com/en-us/microsoft-viva/topics/overview

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 11–19
November 7–11, 2021. ©2021 Association for Computational Linguistics

fresh without the need of any customized intervention. At the core of our knowledge base are entities mined from documents that are of interest to the enterprise, such as product, organization and project. These entities are loosely referred to as topics to the end users (not to be confused with topic modeling in NLP). The knowledge base is a collection of "topic cards" with rich information: properties that help users understand the topic (such as alternative names, descriptions, and related documents), or enable users to connect with people who might be knowledgeable about the topic (related people) or explore related topics.

The contributions of this work are as follows:

- We demonstrate a system in production that performs knowledge mining in large scale: hundreds of millions of documents, thousands of organizations.
- We apply state-of-the-art deep learning models in two NLP tasks named entity recognition (NER) and definition extraction. We discuss the challenges and how we improve our system to reach the desired performance.

2 System description

The overall system architecture is depicted in Figure 1. In this section, we discuss at length the knowledge mining system that works "offline". The system works in a semi-streaming mode: whenever there's a document update, the content of the document is sent to the NER and description mining components. The NER component extracts entities then updates information in the topic candidate store. The topic ranker periodically pulls the topic candidates store to select the top N topics. The topic card builder then builds topics cards with various attributes. Note that this is a simplified view of the actual system. For example, there is another component that conflates information from other sources using techniques described in Winn et al. (2019).

2.1 Named entity recognition for enterprise

NER is the typical first step in information extraction (Jurafsky and Martin, 2009, Chapter 22). Based on our study on enterprise customers' demand and an analysis of Bing's Satori knowledge graph, we define 8 entity types that are of interest to the enterprises while covering most of real-world entities. Among them, "person", "organization", "location", "event", and "product" are the common NER types in various public datasets (CoNLL03: Tjong Kim Sang and De Meulder, 2003, OntoNotes: Hovy et al., 2006; WNUT 2017: Derczynski et al., 2017), while "project", "field of study", "creative work" are less common but are also of high interest to enterprises. These 8 types cover about 85% of entities in Bing's Satori knowledge graph. The remaining entities are mainly biological organisms.

Our NER model is based on Transformers with the pretraining-finetuning paradigm (Devlin et al., 2019). State-of-the-art results on several NER benchmarks are achieved with Transformers (Yamada et al., 2020; Li et al., 2020). To make data collection easier, we train our model on public data but apply it to enterprise domain. The distributional shift between training and testing can cause a significant performance drop (Quiñonero-Candela et al., 2008). To measure model's true performance under distributional shift, we construct a test set from actual internal documents within Microsoft. The size of this test set is comparable to CoNLL03 test set (Tjong Kim Sang and De Meulder, 2003).

To mitigate the distributional shift issue, we divide model training into multiple stages, with the first stage training on large amount of automatically annotated data using Wikipedia, which has been shown to help the system generalize better to a new domain (Ni and Florian, 2016). Entities are identified by wikilink, and we use Bing's Satori knowledge graph to find out the corresponding entity type. We selected paragraphs with at least 10 wikilinks, which gives us $\sim$1 million paragraphs. Finally, we use an entity linking tool NEMO (Cucerzan, 2007, 2014) to annotate entities without wikilinks and get $\sim 50\%$ more entities.

The benefit of Wikipedia training data lies in its size, but it comes with low annotation quality. After training on it, we continue training on smaller data with high quality human annotation. In the second stage, we use OntoNotes 5.0[2] data set and mapped their types to our 8 types. This stage is mainly beneficial for the common NER types, but it does not help our additional "project" and "field of study" type. In the last stage, the training data is a combination of a small number of web documents with 8 types annotation (size is $\sim$1000 paragraphs) and CoNLL03 data with "MISC" type being reannotated to one of our 8 types. This last stage of

[2] https://catalog.ldc.upenn.edu/LDC2013T19

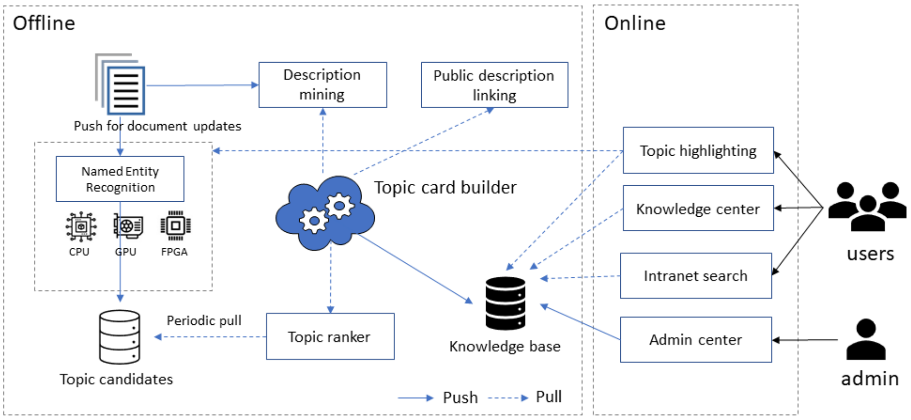
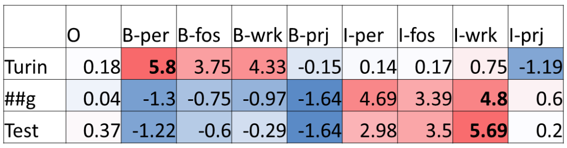


Figure 1: An illustration of the knowledge mining system.



	O	B-per	B-fos	B-wrk	B-prj	I-per	I-fos	I-wrk	I-prj
Turin	0.18	**5.8**	3.75	4.33	-0.15	0.14	0.17	0.75	-1.19
##g	0.04	-1.3	-0.75	-0.97	-1.64	4.69	3.39	**4.8**	0.6
Test	0.37	-1.22	-0.6	-0.29	-1.64	2.98	3.5	**5.69**	0.2

Figure 2: Scores for selected tokens and selected types from the sentence "The history of NLP dates back to the 1950s when Alan Turing proposed a simple test (the "Turing Test") to determine ...". The abbreviations in use are: per for person, fos for field of study, wrk for creative work, and prj for project.

training data is most aligned with our NER type definition.

To illustrate the effect of multistage training as well as additional improvement techniques, we consider BERT-base with cased vocabulary finetuned only on the last stage of training data as a strong baseline. The F1 metrics of our model, baseline, and ablation experiments on our test set from internal documents are shown in Table 1. The baseline 56% F1 is much lower than the reported F1 > 90% on CoNLL03 test set (Devlin et al., 2019), which shows the challenge from the distributional shift (we also test our baseline model on CoNLL03 test set and get F1 > 90%). The most common entity type in our test set is product, which can be more difficult to detect than the most common entity types (person, location, organization) in CoNLL03. Also our test set is noisier than CoNLL03 as internal documents are often less formal than public documents such as newswire articles. Our best model achieved an F1 of 71.1%. In ablation study, multistage training improves F1 by 5.4% from the baseline. We find additional techniques that can robustly improve model performance:

- *Data augmentation*: we find two most useful data augmentation methods out of many methods we have tried. One method is simply lower casing the training data. This method has been shown to increase NER performance on uncased text significantly, and even improve performance on cased text when train and test on different domains (Mayhew et al., 2019). The second method is to replace an entity mention with a randomly selected entity of the same type. This is motivated by our observation that the distribution of entities roughly follows the Zipf's law. Randomly replacing entities can give more weights to tail entities. In combination, data augmentation provides a 0.6% F1 lift.

- *Focal loss*: NER is an imbalanced classification problem as most input tokens are not entities. We test loss functions suitable for imbalanced dataset: Dice Loss (Li et al., 2020), Am-Softmax (Wang et al., 2018), and Focal Loss (Lin et al., 2017). They all provide similar improvement. We report focal loss (hyperparameter gamma=1.6) results here with an additional 1.5% F1 lift.

- *Viterbi decoding*: as there is no hard constraint on the sequence of labels from BERT, the sequence can be invalid under the standard BIO tagging scheme (Lample et al., 2016). Figure 2 shows such an example. The scores for tokens ("Turin", "##g", "Test") give an invalid label sequence of B-per, I-wrk, I-wrk (per stands for person, wrk for creative work) using greedy argmax decoding, which we correct to O, O, O in the baseline setting. We observe that the correct sequence B-wrk, I-wrk, I-wrk has highest sum of scores among

Experiment	Config	F1	P	R
Best model	UniLMv2-large: all techniques	71.1%	72.2%	70.2%
Baseline	BERT base: single last stage	56.0%	54.0%	58.2%
Ablation	BERT base: multi-stage	61.4%	60.7%	62.1%
	+data augmentation	62.0%	60.6%	63.4%
	+focal loss	63.5%	62.7%	64.4%
	+Viterbi decoding	65.4%	65.8%	65.1%

Table 1: NER results on internal test set.

all valid sequences, for example B-per, I-per, I-per. Based on this observation, we use Viterbi algorithm to find the valid path under BIO scheme with the maximum sum of scores. This provides a 1.9% F1 lift. We have also tried adding a CRF layer on top of BERT, training jointly or separately. We do not see additional gain (though CRF layer can further improve 1-layer student model).

- *Bigger and better pretrained model*: BERT is pretrained on English Wikipedia and BookCorpus, which have limited writing styles. Enterprise documents can be more diverse, less formal and noisier. Therefore, pretraining on more diverse corpora may help our task. We switched from BERT base-cased to UniLM v2 large-cased, which is pretrained on additional 144GB corpora including OpenWebText, CC-News, and Stories (Bao et al., 2020). This provides a 5.7% F1 lift.

For production, we distill knowledge from the 24-layer UniLMv2 teacher model into a 3-layer student model, which is initialized from the weights of the first 3 layers of the teacher model (Hinton et al., 2015). We use 1 GB Wikipedia data for distillation. The student model suffers a 5.6% F1 drop. Though not used in production, we experiment continuing distillation with only about 50MB of internal documents. This small amount of data reduces the gap between student and teacher models to 0.9% F1, which suggests the usefulness of using in domain data for knowledge distillation.

Knowledge distillation gives us ∼6x speed up for inferencing a single input sequence on Nvidia V100 GPU with f32 precision. On top of student model, we get another ∼14x speedup by (1) exporting model from Pytorch to ONNX (Ning, 2020), (2) switching from Python to C#, and (3) running inference in f16 precision and batch mode.

2.2 Topic ranker

In the NER step, tens of millions of topic candidates could be detected. The goal of topic ranker is to pick the top tens of thousands most salient topics while reducing the number of noisy topics. We achieve this in two stages by first simply ranking topics by their total number of times being detected by NER (referred to as NER frequency) to produce a short list of topics. Then we rerank the short list by scores from a binary classifier. The classifier is trained to distinguish between good and noisy topics. It uses features such as NER frequency, document frequency, topic-in-title frequency (number of times the topic appears in the document title) and the ratios of these counting features.

This classifier is effective as it uses global statistical information not available during NER stage. For example, the word "Company" could be mislabeled as an organization by the NER model. Although the probability is small, it could still make into the short list as this word appears very often. The classifier would filter it out as the ratio (NER frequency/document frequency) is very small.

Our training set contains 6000 annotated topics detected from about 0.5 million Microsoft internal documents. Using a single feature NER frequency as a baseline, the AUC is 0.54. We train a gradient boosting trees classifier (Ke et al., 2017) using 5-fold cross validation and achieve an average validation AUC 0.67.

In the production system, as the number of topic candidates scales up, the topic ranker could play a more important role as much lower percentage of topics will be selected. To evaluate its true usefulness, we apply the classifier in the end-to-end system to process all Microsoft documents. We randomly sample a subset of topics before and after applying the classifier. We observe a 9% reduction in noisy topics with the classifier.

2.3 Definition mining

A succinct and accurate description is a crucial attribute of a topic. Such descriptions come from two sources: (1) for some topics such as "field of study" type, their descriptions exist in public knowledge and therefore we retrieve this information from Bing's Satori knowledge graph using an existing context aware ranking service, which is in use for Microsoft Word's Smart Lookup feature; (2) more importantly, we build a description mining pipeline to extract private definitions from enterprise documents. This pipeline consists of the following steps:

1. Split a document into sentences.

2. A deep learning model classifies each sentence into one of 5 categories. Pass a sentence in the "Sufficient Definition" category to the next step.

3. Extract topic from the sentence using a list of patterns, for example: topic is defined as description text.

4. Remove sentences with negative opinion (or sentiment) based on lexicon match. We use the Opinion Lexicon from Hu and Liu (2004).

A large corpus contains definition-like sentences with a wide range of ambiguity beyond a binary classification task can capture. Therefore, we make the task more granular and define 5 categories most common in enterprise domain: Sufficient, Informational, Referential, Personal and Non- definitions. Detailed schema is included in the Appendix.

To collect training data, we need to first collect sentences with a relative high chance of being a description. In addition, we want to collect more hard negative examples such as opinions (e.g., "Caterpillar 797B is the biggest car I've ever seen.") than easy negative examples. Using query log from Bing, we achieved these two goals: we collect search results for queries that match patterns such as "what is {term}", "define {term}" as the results are highly related to definitions. The search results also have the advantage of being more diverse than a single corpus. From the search results, we create a set of 42,256 annotated sentences, which is referred as public dataset. As we will show, a model trained on the public dataset suffers a significant performance degrade on enterprise documents due to distributional shift. Therefore, we construct a second dataset from our internal documents that have been approved for use after compliance review, which is referred as enterprise dataset. The model trained on the public dataset is used for identifying candidate sentences for annotation during the construction of the enterprise dataset. Using the enterprise dataset involves many compliance restrictions. For example, we need to delete a sentence if its source document is deleted or our access expires; the model is trained within the compliance zone and stays within it. Details for these two datasets are shown in Table 2, which also includes the DEFT corpus for comparison (Spala et al., 2019). Roughly 15% of the data from the two datasets is withhold from training for testing.

Dataset	# of sentences	# of positive
Public dataset	42,256	10,927
Enterprise dataset	58,780	49,017
DEFT (Spala et. al. 2019)	23,746	11,004

Table 2: Datasets for definition classification task.

Model	Train data	Test data	F1/P/R
Bert-base	Public	Public	0.82/0.76/0.89
		Enterprise	0.64/0.55/0.77
BERT-base	Enterprise	Enterprise	0.73/0.68/0.80
BERT-large			0.72/0.70/0.77
UniMLv2-large			0.75/0.71/0.80
Rule based	N/A	Public	0.48/0.40/0.60

Table 3: Results for definition classification.

Similar to our approach in NER, we consider BERT-base (with cased vocabulary) as a strong baseline. First we train BERT-base model on the public dataset. When testing it on public and enterprise datasets, we get F1 results of 0.82 and 0.64 respectively, as shown in Table 3. This performance degradation again exemplifies the challenge from distributional shift. Then we train on the enterprise dataset and compare BERT-base with BERT-large and UniLMv2-large. UniLMv2-large achieves the best result with F1 of 0.75, which may again benefit from the bigger pretraining corpus (Bao et al., 2020). In Table 3, we also add the result from rule-based classification, which directly uses the list of patterns in Step 3 (e.g., "is a", "is defined as", "refer to") to identify definition. It is evaluated as a binary classification task: "Sufficient Definition" vs Others. We get F1 of 0.48 with an even lower precision of 0.40. This shows the necessity of model-based classification in Step 2 in our definition extraction pipeline.

For production, we distill our best model into a much smaller BiLSTM model. The embedding of the BiLSTM is initialized from 50-dimensional Glove vector (Pennington et al., 2014) with a reduced vocabulary size of 0.12 million. The hidden dimension size is 300. We follow similar knowledge distillation approach as in Tang et al. (2019). The student model reaches F1 of 0.72 while achieves about 30x speedup vs. the 24-layer teacher.

2.4 Topic card builder

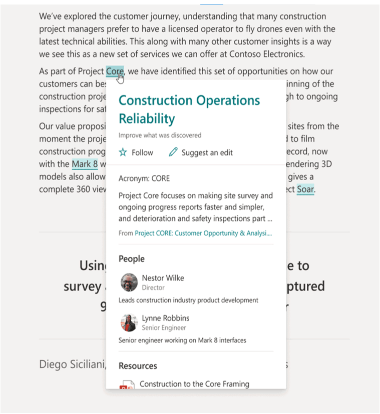
Topic card builder builds topics cards with rich information by aggregating information like definition and acronym from other components. More importantly, it computes the relatedness between topics, documents and users. Using relatedness, it links the top related topics, documents, and users to each topic. By adding related users to a topic, we enable the “expert finding” scenario, which is important for enterprise users to explore expertise. Topic card builder also conflates two topics if their degree of relatedness exceeds a threshold and they pass additional checks to prevent over conflation.

To compute relatedness between any two items, we build a dense embedding vector for each topic, document and user. We apply SVD to decompose the topic-document matrix M into topic and document embeddings, where $M_{i,j}$ is the BM25 of topic i in document j. This is a classical algorithm in collaborative filtering (Koren et al., 2009) and semantic embedding (Bellegarda, 2000; Levy et al., 2015), but the challenge is the size of the matrix M in the j dimension as it can be on the order of tens of millions. We improve a randomized SVD algorithm that iterates on smaller batches of documents so it can solve problem of our scale on a single machine under 8 GB memory limit (Halko et al., 2011). User embedding is represented as the average of embeddings of the documents that the user has authored. Relatedness is computed as the dot product of two embedding vectors. Top K topics and users most related to a given topic are added to the topic card in this manner. For related documents, embedding is used as a recall-oriented step to select candidate documents, and we apply an additional reranking step using additional signals.

To evaluate the overall quality of the system, we conduct human evaluation on the quality of generated topic cards (70K) mined from Microsoft internal documents (40 million). We ask users (Microsoft employees) to judge the overall quality of randomly sub-sampled topic cards by considering all the information. A good topic card means that it has sufficient information to help users understand the topic. In this study, we achieve 85% good rate.

3 Use Cases

The detailed user guide and licensing information can be found on Microsoft Viva Topics website[3]. Here we briefly introduce two ways user can interact with the knowledge base. Figure 3 shows the topic highlighting feature. Topic mentions in documents get automatically linked to the knowledge base. User can hover over the topic mention to see the topic card and click the link in the topic cards to checkout more information. Figure 4 shows an example topic center homepage. The view is personalized as the related topics for a user is presented for the user to confirm. Users can also checkout all the topics from Manage Topics page.


We've explored the customer journey, understanding that many construction project managers prefer to have a licensed operator to fly drones even with the latest technical abilities. This along with many other customer insights is a way we see this as a new set of services we can offer at Contoso Electronics.

As part of Project Core, we have identified this set of opportunities on how our

Construction Operations Reliability

Improve what was discovered

Acronym: CORE

Project Core focuses on making site survey and ongoing progress reports faster and simpler, and deterioration and safety inspections part ...

From Project CORE: Customer Opportunity & Analysi....

People

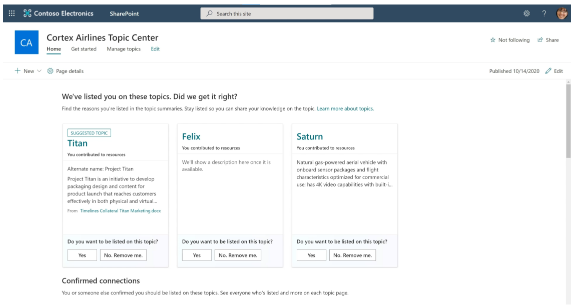
Nestor Wilke
Director
Leads construction industry product development

Lynne Robbins
Senior Engineer
Senior engineer working on Mark 8 interfaces

Resources

Construction to the Core Framing

Figure 3: Snapshot of an example topic card impression in enterprise web document.[4]


Cortex Airlines Topic Center

Published 10/14/2020

We've listed you on these topics. Did we get it right?

Find the reasons you're listed in the topic summaries. Stay listed so you can share your knowledge on the topic. Learn more about topics.

SUGGESTED TOPIC
Titan
You contributed to resources
Alternate name: Project Titan
Project Titan is an initiative to develop packaging design and content for product launch that reaches customers effectively in both physical and virtual...
From Timelines Collateral Titan Marketing.docx
Do you want to be listed on this topic?

Felix
You contributed to resources
We'll show a description here once it is available.
Do you want to be listed on this topic?

Saturn
You contributed to resources
Natural gas-powered aerial vehicle with onboard sensor packages and flight characteristics optimized for commercial use; has 4K video capabilities with built-i...
Do you want to be listed on this topic?

Confirmed connections

You or someone else confirmed you should be listed on these topics. See everyone who's listed and more on each topic page.

Figure 4: Snapshot of a personalized topic center homepage.[4]

4 Conclusion

Organizing resources inside an enterprise into one centralized location facilitates knowledge and expertise sharing and improves productivity. We present a system that automatically constructs a knowledge base from unstructured documents to help enterprises achieving this goal. The system is built upon a combination of deep learning models and classical techniques. We show the challenge of applying NLP models in enterprise domain. We also discuss how we improve the models and the whole system to meet our requirements with detailed experiment results. Finally, we show two typical use cases. We hope our experience can benefit researchers and practitioners in this field.

[3] https://docs.microsoft.com/en-us/microsoft-365/knowledge/

[4] The contents (company name, topic information) are not real internal information but are created for demo purpose.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR.

Jerome R. Bellegarda. 2000. Exploiting latent sematic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1249, New York, New York, USA. ACM Press.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Silviu Cucerzan. 2014. Name entities made obvious: the participation in the ERD 2014 evaluation. In *Proceedings of the first international workshop on Entity recognition & disambiguation - ERD '14*, pages 95–100, New York, New York, USA. ACM Press.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610. ACM.

N Halko, P G Martinsson, and J A Tropp. 2011. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 53(2):217–288.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194(November):28–61.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, volume 65, page 168, New York, New York, USA. ACM Press.

Daniel Jurafsky and James H Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3146–3154.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6256–6261, Hong Kong, China. Association for Computational Linguistics.

Jian Ni and Radu Florian. 2016. Improving multilingual named entity recognition with Wikipedia entity type mapping. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1275–1284, Austin, Texas. Association for Computational Linguistics.

Emma Ning. 2020. Microsoft open sources breakthrough optimizations for transformer inference on gpu and cpu. cloudblogs.microsoft.com.

Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: Lessons and challenges. *Communications of the ACM*, 62(8):36–43.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. 2008. *Dataset Shift in Machine Learning*. MIT Press.

Sasha Spala, Nicholas A. Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. DefT: A corpus for definition extraction in free- and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131, Florence, Italy. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from BERT into simple neural networks. *arXiv*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*, volume 4, pages 142–147, Morristown, NJ, USA. Association for Computational Linguistics.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*, 25(7):926–930.

John Winn, John Guiver, Sam Webster, Yordan Zaykov, Martin Kukla, and Dany Fabian. 2019. Alexandria: Unsupervised High-Precision Knowledge Base Construction using a Probabilistic Program. In *Automated Knowledge Base Construction (AKBC)*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

A Appendix

Category	Description	Example
Sufficient Definition	Can uniquely define and can only define this term.	Statistics is a branch of mathematics dealing with data collection, organization, analysis, interpretation, and presentation.
Informational Definition	Informational but cannot uniquely define this term or can also apply to other terms.	Statistics is a branch of mathematics.
Referential Definition	Is a definition but does not contain the term but instead a reference (“it/this/that”).	This method is used to identifying a hyperplane which separates a positive class from the negative class.
Personal Definition	Associated with the name of a person.	Tom is a Data Scientist at Acme Corporation working on natural language processing.
Non-definition	Does not fall in the other categories. It can be an opinion (hard negative) or not related to definition at all (easy negative).	The Caterpillar 797B is the biggest car I’ve ever seen. "Effective Date" means the date 5th May 2020.

Table 4: Schema for definition categories.