Evaluating Pretrained Transformer Models for Entity Linking in Task-Oriented Dialog

Sai Muralidhar Jayanthi, Varsha Embar, Karthik Raghunathan
MindMeld, Cisco Systems
{saijayan, vembar, ktick}@cisco.com

Abstract

The wide applicability of pretrained transformer models (PTMs) for natural language tasks is well demonstrated, but their ability to comprehend short phrases of text is less explored. To this end, we evaluate different PTMs from the lens of unsupervised Entity Linking in task-oriented dialog across 5 characteristics– *syntactic*, *semantic*, *short-forms*, *numeric* and *phonetic*. Our results demonstrate that several of the PTMs produce sub-par results when compared to traditional techniques, albeit competitive to other neural baselines. We find that some of their shortcomings can be addressed by using PTMs fine-tuned for text-similarity tasks, which illustrate an improved ability in comprehending semantic and syntactic correspondences, as well as some improvements for short-forms, numeric and phonetic variations in entity mentions. We perform qualitative analysis to understand nuances in their predictions and discuss scope for further improvements.[1]

1 Introduction

In task-oriented dialog systems, Entity Linking (EL) is the process of disambiguating a detected entity mention (*aka.* slot) in a user utterance to a canonical entry in a Knowledge Base (KB). EL is a crucial step in building robust dialog systems, especially when dealing with domain-specific entities, *e.g.*, a chatbot for food ordering or a voice assistant for medical assistance.

Popular open-source conversational AI platforms such as DeepPavlov (Burtsev et al., 2018), MindMeld (Raghuvanshi et al., 2018) and Rasa (Bocklisch et al., 2017) maintain a KB of canonical entries, each consisting of a title, optionally with aliases (*i.e.*, alternate usages) for the task of entity linking. Detected entities from user utterances, often with spelling and automatic speech recognition (ASR) errors, are then mapped to those canonical entries through text classification or similarity matching techniques.[2]

Observed Entity	Gold Label	Type
Cabbage Salad pasta with white sauce another position	Cole Slaw Fettuccine Alfredo took on another job	Semantic Equivalence/Domain Knowledge
Super Bowl 48 104 1^{st} street	Super Bowl XLVII 104 First Street	Numeric Equivalence
Great Britian Hungary Sr. DBA rom coms PST Bob	UK HUN senior database administrator romantic comedies Pacific Standard Time Robert	Abbreviations/short forms
doorless croissant ready meaning	Dorlis Prashanth Reddy meeting	Phonetic Similarity/ASR mis-transcriptions
Belgium waffels month to mnth	Belgian waffles monthly	Syntactic Equivalence/ Spelling errors

Figure 1: Different types of matching scenarios observed in Entity Linking task for short spoken/written language texts.

Previous works (Chen et al., 2020; Cao et al., 2021; Broscheit, 2019) have proposed *context-aware* classification techniques for EL, wherein the context surrounding the slots is leveraged to ascertain canonical names. However, such approaches fall short due to **(i)** their reliance on large training/fine-tuning sets and associated annotation costs **(ii)** requirement to re-train the classifiers with every change in KB entries. Alternatively, a more popular paradigm is to model EL as a matching problem by transforming entities into *vectors*, and using a similarity function such as cosine distance to find the closest canonical entry.

EL systems typically rely on textual n-gram features modeled by ranking algorithms such as BM25 (Robertson and Walker, 1994) implemented as part of search engines such as Elasticsearch.[3] To capture semantic similarity within such systems,

[1]Code and re-purposed datasets can be found at https://github.com/murali1996/el_tod

[2]Entity Linking may be clubbed with Entity Recognition or is a standalone component of the NLP pipeline, the latter is used in this work for better interpretability.

[3]https://www.elastic.co/blog/practical-bm25

Proceedings of the 18th International Conference on Natural Language Processing, pages 537–543
Silchar, India. December 16 - 19, 2021. ©2021 NLP Association of India (NLPAI)

one needs to tediously engineer feature sets and collect synonyms or aliases for each KB entry, leading to a lot of manual effort and development cost.

Recently, pretrained word embeddings have had much success in capturing entity correspondences (Francis-Landau et al., 2016; Sun et al., 2015) by addressing aforementioned shortcomings– off-the-shelf usage without reliance on training data and flexibility to expand KBs without retraining. Mudgal et al. (2018) presents a detailed account of different deep learning based representations and modeling choices for the EL task, showing the advantages of using them over traditional systems.

More recently, transformer-based PTMs like BERT (Devlin et al., 2019) have excelled for Entity Linking when entities are in the form of tabular data without much additional *context* (Tracz et al., 2020; Teong et al., 2020; Li et al., 2020; Mudgal et al., 2018). However, their ability to understand nuances in linking short spans of free-form text is not thoroughly tested, especially for domain-specific entities with minimal context.

In this work, we investigate and analyze how different PTMs behave in such settings, when compared to widely adopted neural and non-neural models (**§ 2**). To probe model behaviours on examples with different characteristics, we curate and benchmark evaluation datasets of various sizes that each contain a subset of those characteristics (**§ 3**). Lastly, we present qualitative as well as quantitative analysis of the predictions of various models, which shows that while pretrained models fine-tuned for text-similarity tasks perform the best overall, there is room for improvement (**§ 4**).

2 Models

In this section, we provide a brief detail of the different pretrained transformer models (PTMs) as well as the 5 baseline models (3 neural and 2 non-neural) used in our benchmarking process. We categorize PTMs under consideration into 4 different types to understand the usefulness of different pre-training strategies, number of parameters and inference times. We adopt the model nomenclature from Huggingface[4] (Wolf et al., 2020) and refer the reader to Rogers et al. (2020) and Qiu et al. (2020) for more comprehensive account on these different types of PTMs and their utility.

[4]https://huggingface.co/models

We categorize the PTMs as follows:

Type-I Pretrained general-purpose transformer language models which are *base*-sized. These include *bert-base-cased* (Devlin et al., 2019), *roberta-base* (Liu et al., 2019) and *mpnet-base* (Song et al., 2020).

Type-II Parameters-reduced models which are also trained for language modeling tasks through different parameter reduction techniques. These include *albert-base-v2* (Lan et al., 2020), *distilbert-base-cased* (Sanh et al., 2019), *distilroberta-base* (Sanh et al., 2019), and *MiniLM-L6-uncased* (Wang et al., 2020).

Type-III Reimers and Gurevych (2019) fine-tuned some of the Type-I and Type-II models on a variety of datasets annotated for textual similarity tasks[5]. We select their *all-** models which were fine-tuned with more than 1 billion textual pairs and were designed as general purpose textual similarity models. These include *all-distilroberta-v1*, *all-mpnet-base-v2* and *all-MiniLM-L6-v2*.

Type-IV Dynamic quantization can reduce the size of the model while only having a limited implication on accuracy. We use Pytorch's (Paszke et al., 2019) dynamic quantization functionality[6] to obtain the quantized versions of the following models: *all-mpnet-base-v2* and *all-MiniLM-L6-v2*.

In addition to the pretrained language models based on transformer architecture, we also benchmark PTMs based on other neural architectures. Specifically, we consider the following 3 neural models as baselines– (1) FASTTEXT (Bojanowski et al., 2017), (2) FLAIR (Akbik et al., 2019), and (3) ELMO (Peters et al., 2018).

FASTTEXT consists of continuous distributed word representations trained on large unlabeled corpora for many natural language processing tasks. It represents each word as the sum of its character n-grams. Compared to FLAIR and ELMO, this model has a shallower network and is pretrained similar to Mikolov et al. (2013)'s skipgram model with negative sampling. In our benchmarking, we use the 300-dimension English model.[7]

[5]https://www.sbert.net/docs/pretrained_models.html

[6]https://pytorch.org/dynamic_quantization_bert_tutorial.html

[7]https://github.com/facebookresearch/fastText/crawl-vectors.md

FLAIR is a LSTM based pretrained character language model (Hochreiter and Schmidhuber, 1997), trained to produce a novel type of word embedding also known as *contextual string embeddings*. It is trained without any explicit notion of words and hence can represent even out-of-vocabulary (OOV) words similar to FASTTEXT. In our experiments, we use word representations concatenated from their *news-forward* and *news-backward* models leading to 4096-dimensional vectors.[8]

ELMO is a deep contextualized bidirectional word representation produced by pretrained LSTMs. In our experiments, we use the *base* model and concatenate all three ELMo layers leading to 3072-dimensional vectors.[9]

We compare all the above neural models with two non-neural baselines which are popularly adopted for the task at hand– (1) TFIDF vectorizer[10] and (2) BM25, both using word & character n-grams upto 5-gram.

For all models except BM25, we use cosine similarity as the scoring function. For every pretrained model, we use mean pooled representation of all (sub-)words in a given entity text as its final representation.[11]

3 Datasets

We utilize both in-house and publicly available corpora to curate datasets in English for the Entity Linking task– MindMeld Blueprints dataset[12](MM_BP) along with word and character level misspelled versions of this data (MM_BP-WORD and MM_BP-CHAR), re-purposed open-domain QA datasets like ComplexWebQuestions (COMPLWQ) (Talmor and Berant, 2018) and MKQA (MKQA) (Longpre et al., 2020), acronym identification dataset (ACRI) (Veyseh et al., 2020), and an in-house dataset of ASR mis-transcriptions for person names (ASR-MIS). More details on the dataset curation process is provided in Appendix A.

To probe model behaviours further, we manually annotate 1.3K queries pooled from all of these datasets into our 5 predefined categories as follows (with their sample sizes) – SEMANTIC (#294), SYNTACTIC (#408), SHORT-FORMS (#310), NUMERALS (#125) and PHONETICS (#200). Examples from these sets are presented in Figure 1.

We use Precision@1 (P@1) and Precision@5 (P@5) as our benchmarking metrics and conduct all our experiments using the publicly available MindMeld framework[13]. Unless otherwise stated, we do not include any aliases alongside canonical titles for matching KB entries and utilize all known aliases as our test queries. We disregard any canonical descriptions as they are not always available and procuring them may have significant annotation costs.

4 Results & Analysis

Table 1 present the results of different models across our curated datasets. We observe that on average, Type-I & Type-II models perform poorly compared to the baselines by atleast 30% P@1. However, Type-III & Type-IV models, fine-tuned to find similar sentence pairs, perform superior to our baselines by 5-13%, showcasing the usefulness of such tuning strategies even to short texts. We further observe that the parameter-reduced models generally perform better than the *base* models. Almost all PTMs perform poorly on abbreviations and also fail to beat the BM25 baseline on the phonetic matching dataset. While we believe that these two datasets are quite challenging to the PTMs as their training processes do not include any related objectives, the superior performance of Type-III models compared to Type-I and Type-II is quite encouraging. On misspelled versions of the datasets, Type-III & Type-IV models still perform better than others. However, their precision falls short by at least 10% absolute indicating scope for improvement.

4.1 Qualitative analysis

Figure 2 shows the performance of different models on the 5 different categories of data without and with aliases in the KB. We perform a manual inspection of the results across the 5 categories with 3 different models: baseline BM25 model, Type-I *bert-base-cased* (BERT) and Type-IV *all-mpnet-base-v2-quantized* (MPNET-Q).

[8]https://github.com/flairNLP/flair/FLAIR_EMBEDDINGS.md
[9]https://github.com/allenai/bilm-tf
[10]https://scikit-learn.org/sklearn-TFIDF
[11]Different pretrained models have different tokenization strategies and we leave any analysis on the effect of tokenization to future work.
[12]https://github.com/CiscoDevNet/mindmeld-blueprints
[13]https://github.com/cisco/mindmeld

Results for Entity Linking (Precision@1 / Precision@5)

		MM_BP	COMPLWQ	MKQA	ACRI	ASR-MIS	*Avg.*	MM_BP-WORD *before*	MM_BP-WORD *after*	MM_BP-CHAR *before*	MM_BP-CHAR *after*
Baselines	BM25	49.5 / 52.6	55.1 / 66.0	54.2 / 59.6	1.1 / 1.5	**52.3** / 66.0	42.4 / 49.2	41.9 / 54.9	33.5 / 41.0	46.5 / 60.8	**37.6** / 48.9
	TFIDF	66.7 / 88.3	55.7 / 69.9	67.3 / 84.0	1.0 / 2.1	39.4 / **66.3**	46.0 / 62.1	41.1 / 87.5	**35.3** / 86.3	43.3 / 87.6	36.2 / **87.7**
	FLAIR	44.1 / 81.7	32.8 / 40.2	19.7 / 23.8	0.1 / 0.3	12.9 / 20	21.9 / 33.2	17.1 / 81.9	11.4 / 79.4	20.4 / 82.3	12.1 / 75.3
	FASTTEXT	60.9 / 89.8	37.1 / 47.9	24.4 / 30.7	6.0 / 11.5	4.0 / 7.4	26.5 / 37.5	26.0 / 87.4	13.5 / 82.4	29.4 / 88.7	13.5 / 76.1
	ELMO	57.7 / 84.6	46.6 / 58.3	26.6 / 32.6	1.1 / 1.9	6.4 / 10.4	27.7 / 37.6	17.5 / 86.3	10.4 / 81.5	21.6 / 83.3	10.9 / 75.1
Type-I	*bert-base-cased*	45.9 / 79.4	40.2 / 49.2	27.2 / 34.3	0.6 / 1.2	5.7 / 8.9	23.9 / 34.6	15.4 / 82.0	8.3 / 75.8	18.9 / 78.2	9.6 / 73.7
	roberta-base	59.0 / 76.7	43.9 / 40.4	38.5 / 30.7	1.7 / 0.9	13.1 / 9.3	31.2 / 31.6	27.4 / 78.8	13.5 / 75.3	29.4 / 77.1	17.2 / 72.3
	mpnet-base	31.1 / 74.3	25.2 / 30.1	21.4 / 23.0	0.4 / 0.6	12.3 / 14.0	18.1 / 28.4	21.9 / 77.6	7.1 / 73.1	22.2 / 74.2	8.3 / 69.2
Type-II	*albert-base-v2*	39.1 / 77.8	22.9 / 28.5	17.4 / 19.2	0.1 / 0.3	5.8 / 7.8	17.1 / 26.7	18.2 / 79.9	6.6 / 74.4	18.1 / 78.3	7.0 / 71.7
	distilbert-base-cased	52.8 / 81.1	29.5 / 33.3	37.7 / 41.0	0.6 / 0.7	6.5 / 9.5	25.4 / 33.1	15.4 / 83.9	9.9 / 74.6	19.0 / 80.4	11.1 / 72.8
	distilroberta-base	64.6 / 77.7	39.6 / 31.9	36.1 / 26.5	1.5 / 0.7	10.6 / 6.5	30.5 / 28.7	25.0 / 80.8	14.0 / 74.2	26.2 / 76.9	16.3 / 73.3
	MiniLM-L6-uncased	57.5 / 82.7	33.7 / 38.0	36.8 / 38.9	0.5 / 0.8	16.9 / 18.3	29.1 / 35.8	29.5 / 85.6	9.8 / 75.6	30.6 / 81.4	11.4 / 73.8
Type-III	*all-distilroberta-v1*	72.3 / 91.8	62.5 / 72.1	59.9 / 76.4	**11.2 / 23.3**	42.4 / 62.6	49.7 / 65.2	42.2 / 91.7	24.2 / **87.3**	44.3 / 91.4	32.2 / 87.5
	all-mpnet-base-v2	75.8 / 91.4	62.3 / 72.1	57.6 / 73.3	6.0 / 11.4	43.6 / 57.5	49.1 / 61.1	44.8 / **92.4**	21.6 / 85.4	46.3 / 90.9	27.5 / 86.3
	all-MiniLM-L6-v2	74.6 / 91.7	62.0 / 71.8	68.2 / 80.1	4.7 / 8.2	44.9 / 59.2	50.9 / 62.2	**45.5** / 91.4	21.5 / 82.9	47.5 / 92.0	27.4 / 84.8
Type-IV	*all-mpnet-base-v2* (Q)	**79.4 / 93.3**	**63.2 / 72.4**	**75.5 / 84.1**	5.5 / 10.6	43.5 / 59.5	**53.4 / 64.0**	**45.5 / 92.4**	22.1 / 85.4	46.7 /**92.5**	28.4 / 85.8
	all-MiniLM-L6-v2 (Q)	73.0 / 91.0	61.2 / 70.9	67.3 / 79.6	4.1 / 7.4	42.3 / 58.1	49.6 / 61.4	44.8 / 91.3	20.9 / 83.0	**46.8** / 90.9	26.5 / 82.3

Table 1: Evaluation of different pretrained transformer models across different datasets (§ 3). The *Avg.* column reports mean precision across different datasets. Marked in **bold** are the best scores & in underline are second best.

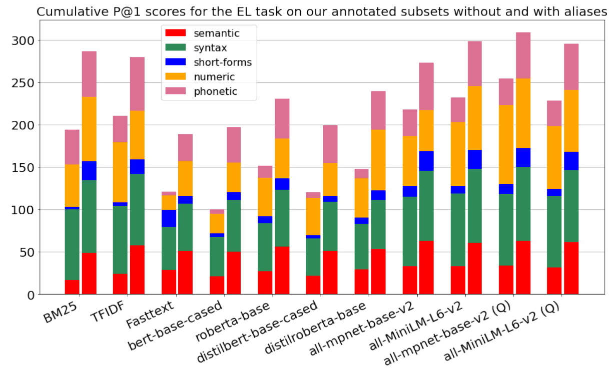

Figure 2: EL results on annotated subset of 1.3K test queries, annotated across 5 matching criterion. For each model, the first bar corresponds to the scenario with KBs containing only canonical names whereas for the second, KBs contain aliases in addition for disambiguating test queries.

4.1.1 Syntactic Matches

Syntactic matches refer to cases when the query and its matching canonical form have slight textual variations or spelling errors. The baseline TFIDF and BM25 models are well equipped to handle such differences and perform on-par and in some cases, better than the other models. Between the BERT and MPNET-Q models, the latter handles syntactic differences better than the former by favouring more word overlaps.

```
Query: John Jr.
BM25: John F. Kennedy Jr.
BERT: Michael Joseph Jackson, Jr.
MPNET-Q: John Warner

Query: mammoth pizza
BM25: Wham, Bam, Thank You Mammoth
BERT: Pizzawich
MPNET-Q: Fresco Pizza

Query: Hindi
BM25: Hindi Language
BERT: India
MPNET-Q: Hindi Language
```

4.1.2 Semantic Matches

The baseline BM25 system relies heavily on aliases to handle queries that are semantically equivalent to one of the canonical names in the KB. In their absence, the model performs poorly in this category. In contrast, the transformer models are better suited to handle these queries. We notice 2 trends in the BERT and MPNET-Q models:
While BERT tends to predict related words, they are not always semantically equivalent.

```
Query: Instrumentalist
BERT: Singer
MPNET-Q: Musician

Query: most recently released
BERT: popular
MPNET-Q: latest

Query: totalled
BERT: count
MPNET-Q: sum
```

In addition, the BERT system tends to rank antonyms higher.

```
Query: min
BERT: highest
MPNET-Q: lowest

Query: hilarious
BERT: erotic
MPNET-Q: comedy

Query: resigned
BERT: active
MPNET-Q: voluntarily terminated
```

4.1.3 Abbreviations & Short Forms

All models perform poorly on abbreviations and short forms. BM25 relies on character n-grams to match shortened sub-strings of entities, but fails on acronyms. MPNET-Q is able to identify acronyms of popular entities like universities, countries, etc., perhaps as a result of the fine-tuning phase.

```
Query: PSU Football
BM25: Football
BERT: UD Arena
MPNET-Q: Penn State Nittany Lions
         football

Query: Mla
BM25: Mlabri Language
BERT: lo
MPNET-Q: Mlabri Language

Query: USSR
BM25: (no result)
BERT: Czechoslovakia
MPNET-Q: Soviet Union
```

4.1.4 Numeric Matches

Among the three systems, BERT performs the worst with numeric entities. It does not handle different numeric representations of the same entity well, leading to random predictions. Fuzzy character matching ensures that BM25 system handles different formats well as long as most of the characters match. MPNET-Q model handles changes in numeric formats the best even when compared against its full model, with P@1 of 92.8.

```
Query: 90's
BM25: 1990s
BERT: 2010s
MPNET-Q: 1990s

Query: 5th Avenue
BM25: 12th Avenue
BERT: 12th Avenue
MPNET-Q: 45 Fifth Avenue

Query: 1775 April 19
BM25: april 1986
BERT: 1875-09
MPNET-Q: 1775-04-19
```

4.1.5 Phonetic Matches

Often, ASR systems mis-transcribe uncommon words into more common, phonetically similar words. This category tests whether the models are robust to such errors. While the performance of all the models are lacking, BM25 qualitatively provides explainable results due to its reliance on textual similarities when compared to predictions of the PTMs. Typically, EL systems are evaluated on queries that test the models' abilities to match the 4 categories mentioned above. Given the popularity of conversational agents with a speech interface, probing EL models for their phonetic matching capabilities is important.

```
Query: this loud (Liz Laub)
BM25: Cloud Hu
BERT: Kevin Upright
MPNET-Q: Riley Rant

Query: Yale sushi (Xiaoxue Shi)
BM25: Sakshi Alekar
BERT: Joshua Frattarola
MPNET-Q: Sammy Su
```

5 Conclusion

Given the success of PTMs for various NLP applications (Rogers et al., 2020), we evaluate the ability of these models to understand short spans of text for unsupervised entity linking in task-oriented dialog systems by curating a large dataset and comparing their results against traditional n-gram systems. We further analyze the performance of these models across 5 different characteristics- syntactic, semantic, abbreviations & short-forms, numeric and phonetic matches. Our results demonstrate that these models, when fine-tuned on a semantic similarity task, comprehend syntactic and semantic differences in short phrases better than their other variants. However, their performance is lacking - particularly for abbreviations and queries with speech recognition errors - with the best performing models averaging at 53.4% P@1 and 64.0% P@5 across the different datasets. For future work, with the goal of creating a generic model for the unsupervised EL task, we plan to improve these models through task-adaptive fine-tuning techniques with our curated datasets.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Tom Bocklisch, Joe Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *ArXiv*, abs/1712.05181.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on*

Computational Natural Language Learning (CoNLL), pages 677–685, Hong Kong, China. Association for Computational Linguistics.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Haotian Chen, Xi Li, Andrej Zukov Gregoric, and Sahil Wadhwa. 2020. Contextualized end-to-end neural entity linking. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 637–642, Suzhou, China. 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.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261, San Diego, California. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.*, 14(1):50–60.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2020. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, A. Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and V. Raghavendra. 2018. Deep learning for entity matching: A design space exploration. *Proceedings of the 2018 International Conference on Management of Data*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *ArXiv*, abs/2003.08271.

Arushi Raghuvanshi, Lucien Carroll, and Karthik Raghunathan. 2018. Developing production-level conversational interfaces with shallow semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 157–162, Brussels, Belgium. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

S. E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval, SIGIR '94, page 232–241, Berlin, Heidelberg. Springer-Verlag.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *ArXiv*, abs/2004.09297.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1333–1339. AAAI Press.

A. Talmor and J. Berant. 2018. The web as a knowledge-base for answering complex questions. In *North American Association for Computational Linguistics (NAACL)*.

Kai-Sheng Teong, Lay-Ki Soon, and Tin Tin Su. 2020. *Schema-Agnostic Entity Matching Using Pre-Trained Language Models*, page 2241–2244. Association for Computing Machinery, New York, NY, USA.

Janusz Tracz, Piotr Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. 2020. Bert-based similarity learning for product matching. In *ECOMNLP*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, and Thien Huu Nguyen. 2020. What Does This Acronym Mean? Introducing a New Dataset for Acronym Identification and Disambiguation. In *Proceedings of COLING*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.