Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2022"

Moscow, June 15–18, 2022

Lightweight and accurate system for entity extraction and linking

Evseev D. A.
Moscow Institute of Physics and Technology
Dolgoprudny, Russia
dmitrij.euseew@yandex.ru

Abstract

Entity extraction and linking components in dialogue assistants should meet the requirements of low resource consumption and high accuracy. In this paper we present lightweight system which extracts entity mentions from the text and finds corresponding Wikidata ids and Wikipedia pages links. Entity extraction and linking is performed into the following steps: extraction of entity substrings from the text, retrieval of candidate entities from Wikidata knowledge base and entity disambiguation. Entity extraction is based on RoBERTa-tiny model for token classification. Extracted substrings are classified into 42 fine-grained tags for filtering of candidate entities. Candidate entities are ranked by number of connections of candidate entities in the text in Wikidata knowledge graph. The proposed system outperforms on WNED-WIKI other lightweight solutions, such as REL and OpenTapioca. The system supports easy adding new Wikidata entities to the database and using other knowledge bases for entity linking.

Keywords: entity extraction, entity linking, entity disambiguation, knowledge base
DOI: 10.28995/2075-7182-2022-21-176-184

Легкая и точная система для извлечения сущностей и связывания с базой знаний

Евсеев Д. А.
Московский физико-технический институт
Долгопрудный, Россия
dmitrij.euseew@yandex.ru

Аннотация

Компоненты для извлечения сущностей и связывания с базой знаний в диалоговом ассистенте должны отвечать таким требованиям, как низкое потребление памяти, а также высокая точность. В данной статье описывается система, которая извлекает сущности из текста и находит для них соответствующие ids в Wikidata и ссылки на страницы Википедии. Извлечение и связывание сущностей происходит в несколько этапов: извлечение подстрок с сущностями из текста, извлечение возможных сущностей из базы знаний Wikidata и устранение неоднозначности сущностей. Компонент для извлечения сущностей основан на RoBERTa-small для классификации токенов. Извлеченные подстроки классифицируются на 42 класса для фильтрации возможных сущностей. Возможные сущности в тексте сортируются по числу связей с использованием графа знаний Wikidata. Предлагаемая система превосходит на датасете WNED-WIKI другие системы с низким потреблением ресурсов, такие как REL и OpenTapioca. Система поддерживает добавление новых сущностей Wikidata в базу данных, а также использование других баз знаний для связывания сущностей.

Ключевые слова: извлечение сущностей, связывание сущностей с базой знаний, устранение неоднозначности сущностей, база знаний

1 Introduction

Entity Linking is the task of identifying an entity mention in unstructured text and establishing a link to an entry in a knowledge base (Sevgili et al., 2021). In dialogue assistants entity linking is a key component for natural language understanding, because entities in the utterance can help to detect user's intention to change the topic and facts from the knowledge base extracted for detected entities can be used for generation of meaningful response.

For parallel dialogue interaction with multiple users entity linking system in a dialogue assistant should be deployed in many replicas, so one of the requirements to EL system is low resource consumption. State-of-the-art entity linking systems are based on large pretrained Transformers (De Cao et al., 2020) or store entities inverted index in RAM (Wu et al., 2019). Lightweight solutions, which store entity embeddings in SQLite database (van Hulst et al., 2020) or use Wikidata (Vrandečić and Krötzsch, 2014) knowledge graph for entity disambiguation (Delpeuch, 2019), stored in Solr[1] index, show low accuracy of entity linking.

In this paper we present lightweight (which can be deployed on an average laptop or desktop machine and does not need much RAM and GPU) and fast entity linking system which can be used in dialogue assistants. The system consists of the following components: identifying entity mention in text, retrieve of candidate entities from the knowledge base, entity mention classifier by types and entity disambiguation using Wikidata knowledge graph and Wikipedia hyperlinks graph. RoBERTa-tiny (Liu et al., 2019) model is used for token classification into three classes: beginning of the entity mention, inside the entity mention and tokens which do not belong to any entity. Detected mentions are classified into 42 tags according to Wikidata entity types with another RoBERTa-tiny model. Candidate entities for the mentions are retrieved from the inverted index in SQLite database with FTS5 extension which supports full text search by entity mentions. For training of RoBERTa model we preprocessed Wikipedia pages with hyperlinks to obtain a dataset of paragraphs annotated with entity mentions and corresponding classes. After filtering we find connections of candidate entities for a mention with candidate entities for other mentions using the knowledge graph. The knowledge graph is stored in the same SQLite database as inverted index which is not loaded into RAM. The proposed system outperforms on WNED-WIKI (Petroni et al., 2020) OpenTapioca and REL. The system does not need pretrained entity embeddings which results in easy adding of new Wikidata entities into the database without need to retrain the models. The system supports entity linking over other knowledge bases provided that the tags of entity type classification model were mapped to knowledge base types.

2 Related work

TagME (Ferragina and Scaiella, 2011) is one of the first entity linking systems, which finds Wikipedia page links for entity mentions in text and uses Wikipedia hyperlinks graph for entity disambiguation. Further improvement of entity linking systems was connected with neural network architectures. In the work of (Ganea and Hofmann, 2017) candidate entities are ranked by bilinear form of entity embedding x_e and embeddings of tokens x_w of K-word local context $c = \{w_1, ..., w_K\}$ (1):

$$\psi(e, c) = \sum_{w \in c} \beta(w) e_w^T B x_w, \tag{1}$$

Global disambiguation, exploiting document-level coherence of entities is performed with CRF-based model. In the system (Le and Titov, 2018) bilinear form is calculated between embeddings of pairs of entities for global disambiguation. In (Le and Titov, 2019) the dataset for training of the model (Le and Titov, 2018) was extended with unlabeled texts with extracted mentions. Candidate entities for the mentions were scored by collective agreement using Wikipedia hyperlinks graph and the entity with the highest score was considered as an answer. In REL (van Hulst et al., 2020) entity disambiguation is based on calculation of bilinear form between entity and context embeddings and entity embeddings for different mentions, the same as in (Le and Titov, 2018). REL system is lightweight because it uses SQLite database for storing entity embeddings. In the approach of (Martins et al., 2019) LSTM is used to extract entity mentions and obtain context embeddings.

In (Kolitsas et al., 2018) all possible n-grams in the sentence were considered as mentions. Entity disambiguation is performed by dot products of candidate entity embeddings and mention embeddings,

[1]https://solr.apache.org/

obtained with LSTM with attention.

Every entity in the knowledge base has the type, (in Wikidata it is defined with the relation P31, "instance of", for example, <Moscow, instance of, city>). In (Raiman and Raiman, 2018) entity types are used for filtering of candidate entities. The document tokens are fed into BiLSTM to obtain mention embeddings, which are fed into dense layer for classification into classes corresponding to types.

In OpenTapioca system (Delpeuch, 2019) candidate entities are ranked by the popularity which is calculated by a log-linear combination of number of statements n_e of entity entity e, site links s_e and its PageRank $r(e)$. Global disambiguation is performed with similarity metrics $s(e, e')$ (the probability that two such one-step random walks starting from e and e' end up on the same item), which are combined using the Markov chain to obtain the score for each entity.

BLINK (Ledell Wu, 2020) retrieves candidate entities from Faiss index of description embeddings. Top N candidate entities descriptions are re-ranked with cross-encoder: the text with entity mention and description of every entity, separated with [SEP]-token, are fed into BERT and dense layer on top of [CLS] hidden state is used for classification into two classes: 1 - entity description corresponds to the mention , 0 - otherwise.

GENRE entity linking system (De Cao et al., 2020) is based on generative model (pretrained BART (Lewis et al., 2019)). GENRE can function in two modes: entity disambiguation, when the text is fed into the model and it generates the text annotated with Wikipedia page links in place of entity mentions, and entity linking, when the entity mention is marked with special token and the model generates the page title.

ExtEnD (Barba et al., 2022) system solves entity disambiguation task the same way as extractive question answering systems. ExtEnD is based on Longformer (Beltagy et al., 2020) which takes as input text with entity mention, marked with special tokens, and candidate Wikipedia page titles, separated with special tokens. The model is trained to find spans of the correct page title.

3 System for entity extraction and linking

The proposed entity linking system consists of the following components: identifying entity mentions in text, classification of entity mentions by types, retrieval of candidate entities from the database, disambiguation of candidate entities using Wikipedia hyperlinks graph.

3.1 Entity recognition

Entity recognition is implemented as classification of text tokens into three classes: "B-ENT" for beginning of the entity mention, "I-ENT" for inner part of the mention and "O" for other tokens. Text tokens are fed into pretrained Tranformer (RoBERTa-tiny), Transformer hidden states are fed into dense layer for token classification.

We trained the model on the dataset of preprocessed Wikipedia pages. The process of page annotation includes the following steps:

1. we extracted all hyperlinks from the page with the corresponding mentions $m^h_1, ..., m^h_N$;
2. for the page and every hyperlink h_i on the page we extracted all Wikipedia surface forms $m^s_{i1}, ..., m^s_{iK}$ using the anchor dictionary (the dictionary where a key is a page title and a value is the list of mentions of the page in Wikipedia);
3. we annotate the tokens of hyperlink mentions $m^h_1, ..., m^h_N$ with BIO-markup;
4. we find substrings which correspond to surface forms $m^s_{11}, ..., m^s_{1K}, ..., m^s_{N1}, ..., m^s_{NK}$ and annotate with BIO-markup.

The dataset contains 130K samples in train set and 2K samples in valid set. RoBERTa-tiny, trained on the dataset, achieves F1=83.2 on valid set and F1=82.6 on test set.

Extraction of more or less entities from the text can be controlled with a threshold in token classification model (A.1).

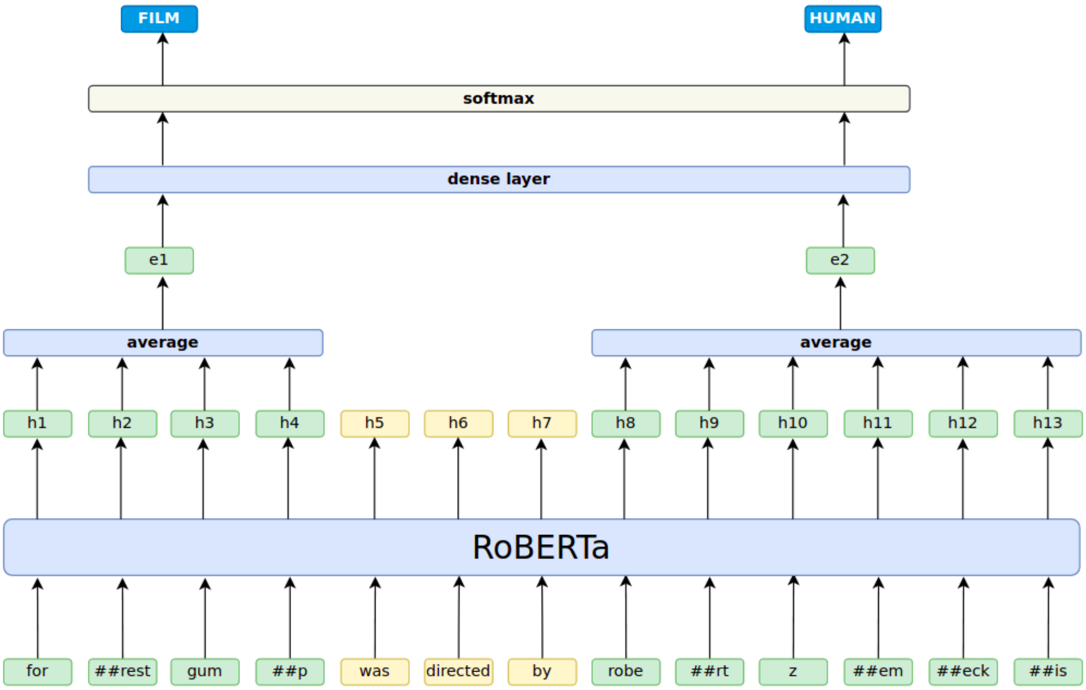

Figure 1: Type classification

3.2 Classification of entity mentions by types

Every entity in Wikidata has the relation P31 ("instance of") or P279 ("subclass of"), for example, <"Forrest Gump", "instance of", "film">. Entity types are useful for entity disambiguation. For example, in the sentence "Forrest Gump was directed by Robert Zemeckis." the type "film" of the mention "Forrest Gump" helps to choose the entity Q134773 ("Forrest Gump", film) instead of entities Q552213 ("Forrest Gump", novel) and Q3077690 ("Forrest Gump", fictional character).

Wikidata contains about 35K types (objects in triplets <entity, P31, type>). We united Wikidata types into 43 types (A.2), for example, Wikidata types "film", "television series", "animated feature film", "feature film", "animated film", "television program" we merged into the type "FILM". All Wikidata entities and corresponding Wikipedia page titles we annotated with these 43 tags.

For classification of entity mentions by types we feed text tokens into Transformer encoder (RoBERTa-tiny in our case). Mention embeddings are obtained by averaging of Transformer hidden states for mention tokens. Mention embeddings are fed into dense layer for classification into 42 classes corresponding to types (Figure 1).

For training of the model we processed paragraphs from Wikipedia pages with hyperlinks. For every hyperlink in the paragraph we found mention spans and the type for the hyperlink page title. We cut long paragraphs to the maximum length of 512 RoBERTa subtokens and left only paragraphs with at least two hyperlinks. The dataset contains 100K in train set and 2K in valid set. The trained model achives F1=79.6 on WNED-WIKI dataset.

3.3 Entity disambiguation with Wikidata graph

In some cases correct entities for the mention are hard to disambiguate based on types. For example, in the sentence "Barcelona defeated Napoli with the score 4:2." the mention "Barcelona" corresponds to the entity Q7156 (FC Barcelona) and in the sentence "Barcelona defeated Valencia BC in the last match." "Barcelona" is Q54893 (FC Barcelona Basquet). We use connections between candidate entities for

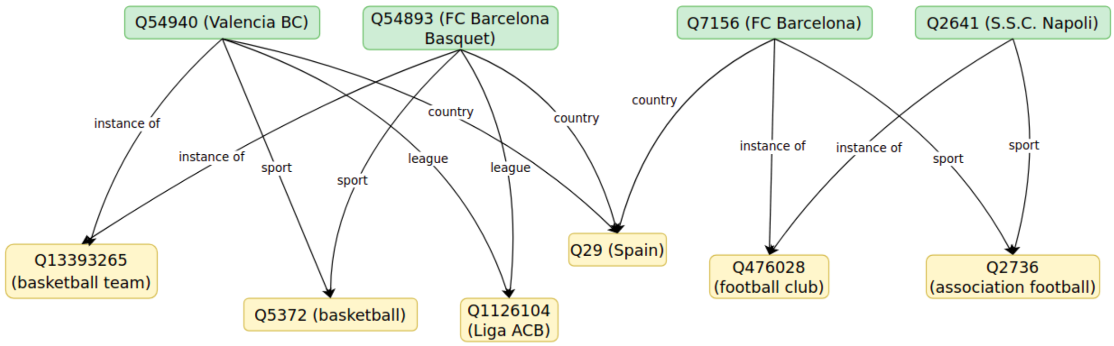

Figure 2: Global disambiguation

different mentions in the text in Wikidata knowledge graph and Wikipedia hyperlinks graph (Figure 2). Entities in Wikidata are mapped to corresponding Wikipedia pages, so we used both relations between entities in Wikidata and edges between pages in Wikipedia in hyperlinks graph. In the former sentence FC Barcelona and "Napoli" (Q2641 (S.S.C. Napoli) are connected with the edge "P31" (isntance of) and the node Q476028 (association football club) and with the edge "P641" (sport) and the node Q2736 (association football).

Disambiguation of entities for mention m in text (for example, entities {"Q7156", "Q54893"} for mention "Barcelona" in the sentence "Barcelona defeated Napoli with the score 4:2.") was inspired by (Usbeck et al., 2014). For each entity $e^i_j \in C^i = \{e^i_1, ..., e^i_{Ni}\}$ for mention m^i we find all entities in Wikidata connected with e^i_j with outgoing edges (the edges in directed graph that begins in e^i_j). In Figure 2 the edge, outgoing from the entity Q7156, connects Q7156 and Q29. We build a graph $G_k = (V_k, E_k)$, where $V_0 = \{C^1, ..., C^N\}$ (candidate entities), $E_0 = \emptyset$, E_1 are edges outgoing from the nodes V_0, V_1 are found as follows (2):

$$V_1 = V_0 \bigcup \{y : \exists x \in V_i \wedge (x, y) \in E_1\} \tag{2}$$

All nodes $x, y \in V_1$ we initialize with authoritative values $x_a = \frac{1}{|V_1|}$ and hub values $x_h = \frac{1}{|V_1|}$ and iterate k times (3):

$$x_a \leftarrow \sum_{(y,x) \in E_1} y_h, y_h \leftarrow \sum_{(y,x) \in E_1} x_a \tag{3}$$

After k iterations all candidate entities $e^i_j \in C^i$ for mention m^i have corresponding values x^i_{aj}, candidate entities are sorted by x^i_{aj}.

4 Evaluation

The proposed entity extraction and linking system was tested on WNED-WIKI dataset. The system outputs three confidences: the Levenshtein distance between the mention (entity substring in text) and Wikidata entity title, the confidence of entity type classification model (Section 3.2) and the score of proximity with other mentions in Wikidata graph (Section 3.3). The final confidence was obtained as linear combination of these confidences and if the confidence is lower than the threshold, the entity mention was considered as not found in Wikidata.

WNED-WIKI dataset contains 6.8K samples with mentions from Wikipedia paragraphs and corresponding page titles. The proposed system outperforms REL (van Hulst et al., 2020) and OpenTapioca (Delpeuch, 2019) on WNED-WIKI (Table 1). OpenTapioca disambiguates candidate entities by the number of connections between entities for different mentions is Wikidata graph. REL is based on ranking of candidate entities by dot products of entity and context embeddings. Global disambiguation in REL is performed by calculation of dot products of candidate entity embeddings for different mentions, but the system does not use explicit information about connections between entities in Wikidata knowedge graph. Our system performs both local disambiguation (filtering of candidate entities by types obtained from type classification model) and global disambiguation by proximity of candidate entities in Wikidata.

GENRE, ExtEnD and BLINK systems achieve high F1 because they are based on powerful methods of page title generation (GENRE), extraction of page title span from the list of candidate titles (ExtEnD) and cross-attention between text and candidate entity description (BLINK) with large pretrained Transformers. GENRE is an encoder-decoder model with two modes:

- taking text with entity mention marked with special tokens as input and generating the page title;
- taking text as input and generating the same text where entity mentions are replaced with page titles.

Generation of page titles in autoregressive way, token-by-token, allows to learn relations between context and entity name.

The main component of ExtEnD system is a Longformer which recieves the text where the entity mention is marked with special tokens, and the list of candidate pages titles. The model is trained to extract the span of correct page title the same way as extractive question answering models. Longformer hidden states are fed into two dense layers, the first defines the probability of the token to be the span start, the second - the span end. Cross-attention in Transformer architecture between page title, entity mention and text tokens leads to effective learning of relationship between page title and context.

BLINK system consists of two components: extraction of candidate entities from Faiss index and re-ranking of entities. At re-ranking step the text with entity mention replaced with special token and candidate entity description are fed into BERT and dense layer on top of CLS-token hidden state is used for classification of the description into two classes: 1 - if the description correponds to the context, 0 - otherwise.

Large pretrained Transformers in GENRE and ExtEnD result in high quality, but using Longformer in ExtEnD leeds to low inference speed. In GENRE prefix tree of 6M Wikipedia pages is loaded to RAM and requires 6.1 Gb. Also, GENRE and ExtEnD does not support zero-shot transfer to other knowledge bases. BLINK system is zero-shot: the entity is defined only by short text description, but the entities index (5.3 M) is loaded into RAM which requires 37.5 Gb. Cross-encoding of text and entity descriptions in BLINK is slower compared with other methods (Table 1) because the input text should be fed into BERT the number of times equal to the number of candidate entities. To obtain memory requirements of the models we launched each of the models on Nvidia DGX-1 server with Tesla P100 GPUs and inferred on WNED-WIKI dataset.

The proposed system shows lower F1 than GENRE, BLINK and ExtEnD on WNED-WIKI, but is fast and much more lightweight and can be used on an average laptop or desktop computer. Our system is based on RoBERTa-tiny for entity extraction and type classification and stores entity inverted index and Wikidata graph in SQLite database (2.5 Gb on disk, 42.9 M rows) which is not loaded into RAM (**??**). Moreover, our system does not need pretraining of entity embeddings and therefore supports easy adding of new Wikidata entity (with one insert query to SQLite database) and transfer to other knowledge bases, provided that the types of entities in the knowledge base were mapped to tags of entity type classification model.

Model	RAM, Gb	GPU, Gb	WNED, micro F1	Inference time, per 1 sample
Our system	1.9	1.4	68.2	0.15
GENRE	9.7	2.8	87.4	0.15
BLINK	37.5	1.1	75.5	0.61
ExtEnD	4.5	2.5	88.8	1.1
REL	2.0	0.95	41.4	0.17
OpenTapioca	4.4	0	26.8	0.21

Table 1: Comparison of the proposed entity linking system with other solutions

To define the contribution of entity linking system components into the metrics, we tested entity linking system on WNED-WIKI in two settings:

- using only entity type classification component for entity disambiguation;
- using both entity type classification and entity disambiguation with Wikidata graph.

In the former setting we achieved micro F1 of 49.8 on WNED-WIKI, in the latter setting - 68.2. The results indicate that connections in Wikidata and Wikipedia between entities in text for different mentions are significant for entity disambiguation and improve the metrics relative to using only entity type classification by about 18 points. For example, in the sample from WNED-WIKI "Towns within the division include Pipers River, Scottsdale, Evandale, Swansea, ..." for the mention "Swansea" the system in setting with using for disambiguation only entity types chooses the wrong entity Q23051 ("Swansea"). Wikidata graph helps to define to correct entity Q986654 ("Swansea, Tasmania"), because most of the locations in the sample text are connected with the entity Q34366 ("Tasmania").

5 Conclusion

In this work, we have described the system for entity extraction and linking. The system performs detection of entity mentions in the text, candidate entities retrieval, entity classification by types with RoBERTa-based model and entity disambiguation using Wikidata knowledge graph. The system is lightweight: entity extraction and type classification components are based on RoBERTa-tiny model, entities inverted index and Wikidata are stored in SQLite database, which is not loaded into RAM. Our system outperforms other lightweight solutions on WNED-WIKI dataset due to combination of local disambiguation based on filtering of candidate entities with type classification component and global disambiguation by proximity of candidate entities in Wikidata knowledge graph.

References

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2022. ExtEnD: Extractive entity disambiguation. // *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Online and Dublin, Ireland, May. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Antonin Delpeuch. 2019. Opentapioca: Lightweight entity linking for wikidata. *arXiv preprint arXiv:1904.09131*.

Paolo Ferragina and Ugo Scaiella. 2011. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software*, 29(1):70–75.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. *arXiv preprint arXiv:1808.07699*.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. *arXiv preprint arXiv:1804.10637*.

Phong Le and Ivan Titov. 2019. Boosting entity linking performance by leveraging unlabeled documents. *arXiv preprint arXiv:1906.01250*.

Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. 2020. Zero-shot entity linking with dense entity retrieval. // *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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 preprint arXiv:1907.11692*.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2019. Joint learning of named entity recognition and entity linking. *arXiv preprint arXiv:1907.08243*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: multilingual entity linking by neural type system evolution. // *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2021. Neural entity linking: A survey of models based on deep learning.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. 2014. Agdistis-graph-based disambiguation of named entities using linked data. // *International semantic web conference*, P 457–471. Springer.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. // *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, P 2197–2200.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

A Appendix

A.1 Confidence threshold for tags in token classification model

The softmax layer in token classification model outputs confidences of every class for the token (B-ENT, I-ENT, 0). We do not follow the strategy of choosing the label L_{ij} with maximal confidence p_{ij} for the token t_i. Instead, we set a threshold and choose the maximum of B-ENT and I-ENT confidences (if it is below the threshold) and O-tag otherwise (4):

$$L_i = \begin{cases} \text{B-ENT}, & p_{i,b-ent} > p_{i,i-ent} \,\&\&\, p_{i,b-ent} > thres \\ \text{I-ENT}, & p_{i,i-ent} > p_{i,b-ent} \,\&\&\, p_{i,i-ent} > thres \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The example of regulation of entity extraction with the threshold (the sample from WNED-WIKI dataset): if the threshold of B-ENT and I-ENT is 0.7, the model extracts from the sentence "Noel Mary Purcell was an Irish rugby union and water polo player." substrings "Noel Mary Purcell" and "Irish", if the threshold is 0.1, the substrings "rugby union" and "water polo" are also extracted as entities.

A.2 Tags for entity classification

Tags of entity type classification model were mapped with types of entities in Wikidata (types are defined with the relation P31 ("instance of"), for example, <"Forrest Gump", "instance of", "film">). The table 2 contains entity tags and corresponding entity types. For example, the tag "RIVER" is mapped to the type Q4022 ("river").

Using this mapping, we found tags for all Wikidata entities. The search of types was recursive (if the entity has the type which does not correspond to any tags, we found the types of the type, and so on till the one of the types matched any tag, the recursion depth was constrained to 10 steps). If no tag was found, the entity was assigned to "MISC" ("miscellaneous") tag.

Entity tag	Wikidata types
film	Q11424, Q5398426, Q29168811 Q24869, Q202866, Q15416
song	Q482994, Q55850593, Q7302866 Q105543609, Q134556
literary work	Q7725634
animal	Q729, Q7377, Q57814795, Q39201
sport team	Q847017, Q12973014
food	Q2095, Q19861951
city	Q7930989
country	Q7275, Q6256
fac	Q12280, Q811979, Q12819564 Q41176, Q1248784 Q34442, Q25631158
event	Q1656682, Q108586636, Q16510064
product	Q431289, Q167270, Q2424752
law	Q3150005, Q93288, Q1864008
language	Q20829075, Q20162172 Q34770, Q33742
nation	Q6266, Q41710, Q81058955 Q33829, Q231002
norp	Q4392985, Q9174, Q110401282 Q5390013, Q7257, Q49447, Q82821
per	Q5
loc	Q1048835, Q15642541, Q486972 Q82794, Q618123
org	Q43229

Entity tag	Wikidata types
work of art	Q838948, Q17537576
academic discipline	Q11862829
type of sport	Q31629
music genre	Q188451
sports season	Q27020041
sports event	Q13406554, Q18608583
county	Q28575
politician	Q82955
actor	Q33999
writer	Q36180, Q28389, Q49757
musician	Q639669, Q177220, Q36834 Q753110, Q488205
athlete	Q2066131, Q18536342
national sports team	Q1194951
river	Q4022
road	Q34442
business	Q4830453, Q891723 Q6881511, Q783794
occupation	Q4164871, Q12737077, Q28640
chemical element	Q11344, Q11173
sports league	Q623109
political party	Q7278
us state	Q35657
association football club	Q476028
championship	Q1344963, Q500834, Q1079023
sports venue	Q1076486

Table 2: Mapping of entity classification tags and Wikidata entity types

A.3 Candidate entities retrieval

Index of entities with corresponding Wikipedia page titles and Wikidata triplets is stored in SQLite database with FTS5 extension. The row in the table with entities contains entity title, entity id in Wikidata, Wikipedia page title, entity tag and string with Wikipedia triplets (in which the entity is the subject) and hyperlinks on corresponding Wikipedia page, separated with tabulation. The size of database is 2.5 Gb on disk, the database contains 42.9 M rows.

For retrieval of candidate entities we execute a query to the database which contains entity substring and top-3 tags, detected with entity type classification model. If the confidence of top-1 tag is lower than the threshold ($thres = 0.4$), "MISC" tag ("miscellaneous") is added to the set of tags in the query.