Efficient, embedding based candidates generation, to improve entities and relations linking

1. Motivation

With the rise of publicly available Knowledge Graphs (KG) came also a need to harvest these large amounts of data in efficient ways to solve interesting problems in the domain of Natural language processing. One problem, in particular, is Question Answering over Knowledge Bases (QAKB). However, one way to solve this problem, is by combining multiple components each responsible for solving a different subtask [10]. This paper we present an efficient approach to solving one of these subtasks, namely Candidates Generation (CG).
Assuming that entities and relations mentioned in a given input sentence are already extracted, we rely on sentence embeddings determine the most probable referred to entities and relations.
Finally, we will evaluate the performance of the presented component as a seperate system, and the improvements it introduces to the downstream subtask of Entity Linking (EL) (refer to Section:4).

2. Related works

Over time, the entity and relation linking problem has attracted a diverse range of solutions. This is due to the fact that EL-systems have been proven to be useful across multiple domains; such as information extraction, biomedical text processing, or semantic parsing and question answering, to name a few[4].
Many of these solutions rely on one of the most relevant text-based EL algorithms[3] [2]. As a first step, a list of candidate entities for the mentions in the document are generated, and in a second step, with the help of some disambiguation system, the best candidates are chosen.
Our work relates to the first step of this algorithm, namely the Candidates Generation (CG). Up until recently, there are three classes of approaches (1)*Surface Form based approach*, (2)*mention expantion and aliases based approach*, and (3)*prior probability computation based approach*[4].
Many systems combine these classes of approaches to generate candidate entities [6, 5, 1].

For example, **AGDISTIS** relies on surface form, by employing text preprocessing techniques such as string normalization, as well as on mention expansion. In **MAG**[1] a context Index is further introduced that relies on Concise Bounded Description (CBD) [1].
More recent approaches, however, [7][9] select candidates based on embeddings generated by language models. In (Wu et al. 2020)[7], for instance, bi-encoders are used to retrieve candidates which are then, along with the mention/context, are encoded in a single transformer. And in a final step a score for each pair is computed.

3. The approach

As mentioned previously, we assume that the entities and relation mentions in the input document are already determined. Thus we start with a sentence and a list of the mentions as input.
The main idea will consist of embedding the input, and looking at the candidate generation problem as a prediction or translation problem. We will try to map the mention's embeddings, which is the embedding of the input with the attention shifted to a specific mention, to the embedding space of the underlying knowledge graph. Then, beginning with the projected vector, we build a list of candidates based on the entities that are the closest to it. An oveview of the described approach can be found in Figure 1
For the embedding of the input document we will rely on a pretrained model (e.g. BERT[2]). As for the KG, we will test our approach on different Embeddings of the DBPedia[3] KG. We will at a later stage assess the results of our CG method on these different embeddings and draw conclusions.
In order to compute a prediction for the mentions, we will make use of the transformer architecture[8], and with the help of the attention layers we will be able to compute a vector in the embedding space of KG for each mention.

4. Evaluation

For the evaluation, we will use separate measurements for the Candidate generation and the Entity disambiguation module.
To evaluate the performance of the Candidate generation process, we will compute recall, or how frequently the actual referred entity occurs in the candidates generated.. Because it is imperative, to achieve good performance on the Disambiguation step, to have the actual referenced entity present in the candidates set.
Moreover, we will compare the performance of an entity linking system [1] before and after introducing the approach described here for candidate generation in the system, by measuring its recall, precision and F1-score.

[1] https://www.w3.org/Submission/CBD/
[2] https://huggingface.co/bert-base-uncased
[3] https://www.dbpedia.org/

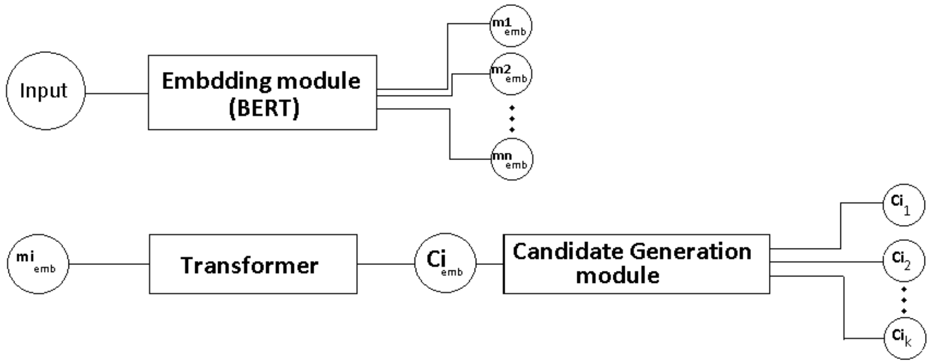
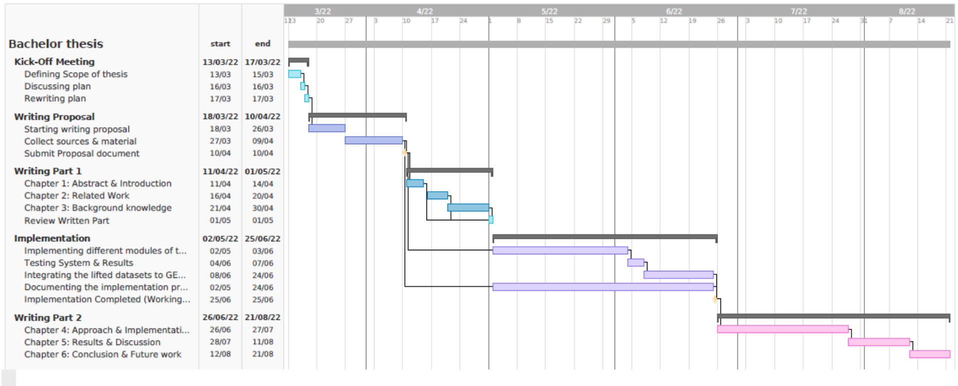


Figure 1: An overview of the presented approach.
The set $M_{emb} = \{m_1, m_2, \ldots, m_n\}$ is the embedding of the mentions in embedding space E
The Transformer then maps each embedding to a vector C_i in the Embedding space of the knowledge base E_{KB}
Finally the Candidate Generation module generates a list of the k closest entities in E_{KB} to the vector C_i

5. Conclusion

A good performance at the Candidate generation step is vital for good results from the overall system. We hope, by combining the mentioned methods above for generating candidate entities and relations to improve the performance of the disambiguation module and the Entity Linking systems in general.

6. Time plan



Bachelor thesis	start	end
Kick-Off Meeting	**13/03/22**	**17/03/22**
Defining Scope of thesis	13/03	15/03
Discussing plan	16/03	16/03
Rewriting plan	17/03	17/03
Writing Proposal	**18/03/22**	**10/04/22**
Starting writing proposal	18/03	26/03
Collect sources & material	27/03	09/04
Submit Proposal document	10/04	10/04
Writing Part 1	**11/04/22**	**01/05/22**
Chapter 1: Abstract & Introduction	11/04	14/04
Chapter 2: Related Work	16/04	20/04
Chapter 3: Background knowledge	21/04	30/04
Review Written Part	01/05	01/05
Implementation	**02/05/22**	**25/06/22**
Implementing different modules of t...	02/05	03/06
Testing System & Results	04/06	07/06
Integrating the lifted datasets to GE...	08/06	24/06
Documenting the implementation pr...	02/05	24/06
Implementation Completed (Working...	25/06	25/06
Writing Part 2	**26/06/22**	**21/08/22**
Chapter 4: Approach & Implementati...	26/06	27/07
Chapter 5: Results & Discussion	28/07	11/08
Chapter 6: Conclusion & Future work	12/08	21/08

References

[1] Diego Moussallem, Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach (2017). arXiv:1707.05288

[2] Parravicini, Alberto and Patra, Rhicheek and Bartolini, Davide and Santambrogio, Marco, Fast and Accurate Entity Linking via Graph Embedding (2019), doi 10.1145/3327964.3328499

[3] Delip Rao, Paul McNamee, and Mark Dredze. 2013. Entity linking: Finding extracted entities in a knowledge base. In Multi-source, multilingual information extraction and summarization. Springer, 93–115.

[4] Neural entity linking: A survey of models based on deep learning, Sevgili, Ozge and Shelmanov, Artem and Arkhipov, Mikhail and Panchenko, Alexander and Biemann, Chris, arXiv preprint arXiv:2006.00575, 2020

[5] Robust Disambiguation of Named Entities in Text, Hoffart, Johannes and Yosef, Mohamed Amir and Bordino, Ilaria and Fürstenau, Hagen and Pinkal, Manfred and Spaniol, Marc and Taneva, Bilyana and Thater, Stefan and Weikum, Gerhard

[6] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS – Graph-based disambiguation of Named Entities using Linked Data. In International Semantic Web Conference (ISWC), pages 457–471. Springer, 2014.

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

[8] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30. arXiv:1706.03762

[9] Increasing Entity Linking upper bound through a more effective Candidate Generation System (2022), Anonymous ACL submission

[10] Lan, Yunshi, et al. ”A survey on complex knowledge base question answering: Methods, challenges and solutions.” arXiv preprint arXiv:2105.11644 (2021).