UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Research Group undefined

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

by
Youssef Ameur

Paderborn, May 4, 2022

Abstract. 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 assess the performance of the provided component as a standalone system as well as its performance as a component of an entity linking system, specifically the enhancements it introduces downstream.

Contents

1

01-Introduction

The task of linking names in free text to referent entities in a knowledge base is known as entity linking. The use of the entity linking task spans over multiple domains (e.g. Information extraction, biomedical text processing, ...) [05]. Here, we focus mainly on how it assists text analysis in comprehending the context of the name in depth by using known entity information. The most recently proposed linking systems are divided into two steps: candidate creation (aka. candidate generation) and candidate ranking (aka. entity disambiguation).

Numerous works have been done to improve the systems that perform the task of entity linking, and many of these works have focused on the second step of the named task [01]. The developed systems have achieved considerable results, e.g. in [02], they achieved an accuracy of 95% on five standard entity disambiguation datasets.
Nonetheless, regardless of how accurate the system is, the first stage of candidate creation is critical to a solid performance, since the absence of the referred entity in the candidate set generated will inevitably lead to an erroneous output.

Although simple approaches that rely on string similarity or Wikipedia anchor-text links have achieved a high recall [ ], these methods are not without their own set of obstacles and issues. We will discuss these in the following chapter in more detail.

The technique we will present in this work bypasses the surface form of any given mention. By relying on word embeddings generated with the help of pre-trained models, we encode the present mentions and their context and use high-density vectors to represent them.
In turn, we will map the resulting vectors to an embedding space that represents the entities contained in the underlying knowledge graph, which entities we are trying to link the mentions to. This strategy is similar to the one described in [03, 04]'s works and others.

In the remaining part of this work, we will proceed as follows:

- First, we will consider some of the approaches adopted to generate candidate entities for mentions, that do not rely on embeddings and we will discuss their challenges and shortcomings. Then we will mention certain methods that inspired this work while discussing the similarities and the differences.
- Second, after giving some background knowledge on the task to be executed, we will explain the inner workings of the system we implemented in detail, and how we went about implementing it.
- Next to last, we present the results of the tests that our system will undergo, in order to measure its performance, followed by an analysis.
- Finally, based on our findings and previous works we will draw conclusions and state any possibilities for improvements and future work.

2

Formal Notations

3

Related Work

4

Approach

5 Comparison

6 Discussion