Using Named Entity Recognition to Improve Accuracy of the DrQA Model

Anonymous { Individual }

Abstract

This paper proposes improvements to the performance of a given baseline model from the DrQA paper (Chen et al., 2017) by using Named Entity Recognition. The NER is typically used as an aid to filter out strings that do not contain the answer (Mollá et al., 2006). My approach is to use NER to contextualize the questions and answers and focus on sections of text that contain the correct answer. I trained and tested this approach on the given SQuAD and SQuAD adversarial data sets on Google Colab and got some improvements over the baseline model.

1 Introduction

Through this course we have experienced different LSTM neural networks to identify the most likely probabilities for the start and end of the answer span in each passage to answer a question. In this project, I explored a different approach to modify this spanning algorithm.

Named entities are sets of elements that are important to understanding the text. Some common entities come from parts of speech (like nouns, verbs, adjectives, etc). The process of identifying named entities in the text is known as Named Entity Recognition (NER) and is a subset of the topic of information Extraction. Some researches have shown that Named Entity Recognition successfully improved different aspects of QA performance.

NER has helped preprocess data to reduce QA input (Noguera et al., 2005), which helps with the efficiency of the QA system. What's more, NER can be used to map question types to entity labels, meaning that certain questions should have an answer that falls into some specific subset of named entity categories (Mendes et al., 2010). The approach that I implemented and tested is based on this idea. In this paper, instead of improving the efficiency of the system, I decided to use NER to directly improve the EM/F1 scores in adversarial settings.

2 Method

2.1 Named Entity Recognition

Named Entity Recognition systems are used in a lot of NLP tasks. In particular, they play an important role in Question-Answering. Named Entity Recognition systems are typically used in question answering systems like AFNER to narrow down the candidate answers which match the semantic category of the selected answer. For example, the answer to the question "which is the capital of the united states", the system identifies the category of the expected answer to be a Location(LOC). Thus, the system will only consider the named entities with the category LOC as answers thereby affecting both the precision and performance of the overall system.

2.2 Named Entity Mapping

The reason I use Named Entity Mapping is that I think it helps me to figure out what type of answer the question was looking for based on keywords. For example, if a question has the word "who" in, it would suggest that the answer could be a person or an organization. With that information, I can capture the type of named entity the question asked for. Based on the specific keywords in the question, I assigned higher probability scores to certain types of named entities in the passage. These question keywords are questioning words that strongly indicate that they are looking for a certain type of named entity. For example, the questions with "where" interrogative, the answers that have a named entity of a category that corresponds to the question's interrogative will have priority. With this idea, I believe it will help improve the performance of the system.

2.3 Evaluation of potential answers

Apart from simply implementing NER into the existing QA system, I also need to consider how to use it to improve the existing probability distribution analysis that the system uses to choose the answer. I created a voting system with probability multipliers that integrate the information from the base QA system and the results of running NER on the question/answers to choose the best answer.

2.4 Implementation

Implementing NER into the base QA system required no changes to the model. Therefore, there are no changes in model.py. Most of the changes were in the `write_predictions` method in the main file, which calculates answers to questions based on the probabilities from the QA system. In the util file, instead of using the `search_span_endpoints` function to get one optimal answer, I created a `top_k_span_endpoints` function that returns a list of k answers. To test this, I changed the call to `search_span_endpoints` in the `write_predictions` method to `top_k_span_endpoints[0]`, which logically should give us the best answer and the same EM/F1 scores when using the given baseline model (`baseline_small_squad.pt`). I found that the evaluation scores were the same (EM: 48.76, F1: 61.13) on SQuAD data sets. Therefore, the function was working as expected.

I also wrote code to implement NER on the potential answers and compare them to the question text in `write_predictions` method. This code takes a start/end index and finds the sentence surrounding it, so we can find named entities in the area around a potential answer span and not forget the context around an answer. I encountered an issue when trying to use the spaCy Python NLP library. I found that spaCy doesn't find the named entities well in lower-cased text. To fix this, I modified the QADataset class in the data file to take a lowercase boolean argument which allowed the method to maintain both lowercase and true cased data sets for usage in the main file. In the `write_predictions` method, I first call the `top_k_span_endpoints` to get the top k answers and also get the question from the true cased dataset. Then I passed this top k answers through my approach for applying NER to improve the system.

My approach included finding the first interrogative in the question (like ”who”) and finding named entities in the answer that are in the proper category to the question. I created a dictionary from interrogatives to their corresponding named entity categories in spaCy. For example, ”who” maps to the categories ”PERSON” and “ORG” (companies, agencies, institutions, etc.), which means that named entities in those categories are more likely to be part of an answer to a ”who” question. The list of our question keywords along with their corresponding named entities in the table below.

Question Keyword	Corresponding Named Entities
who	”PERSON” ”NORP” ”ORG” ”GPE”
where	”FAC” ”ORG” ”GPE” ”LOC” ”EVENT”
when	”DATE” ”TIME”

I then apply a multiplier to the probabilities by using the `apply_multiplier` method that I created in the main file to weigh answers higher if they have the named entities corresponding to the question's interrogative.

3 Results

NER can help analyze the structure of the question and the document and possibly allow us to take steps to identify candidate answers or rule out bad answers. I expected the system to perform better in adversarial settings. Therefore, I trained and tested the system on both the SQuAD (Rajpurkar et al., 2016) and adversarial SQuAD (Jia and Liang, 2017) datasets provided to us. I used the EM(exact match) and F1 score as metrics to evaluate the results. In order to evaluate the system efficiently, I used Google Colab to train and test the datasets. The following tables show the results of the integration of NER into the base QA system.

	EM	F1
Baseline	48.75	61.13
Named Entity Mapping	50.72	62.71

Table 1: SQuAD Dataset.

	EM	F1
Baseline	37.72	47.86
Named Entity Mapping	39.78	49.39

Table 2: Adversarial SQuAD Dataset.

4 Conclusion

From these results, we found that our hypothesis that named entities would improve the accuracy was correct. Our method in integrating NER into the baseline QA system did improve question answering systems in adversarial settings.
Compared to the baseline scores:

- For the SQuAD dataset, the EM score is increased by nearly 2 %, and the The F1 score is increased by 1.5 %.

- For the SQuAD adversarial dataset, the EM score is increased by 2 %, and the F1 score is increased by 1.5 %.

5 Acknowledgements

I would like to thank professor(Greg Durrett) and TAs for this wonderful semester. I would not have gotten through this semester without your hard work and support.

6 Appendix

The table below shows a complete list of named entity types and their descriptions.

Named Entity Type	Description
PERSON	People, including fictional
NORP	Nationalities or religious or political groups
FAC	Buildings, airports, highways bridges, etc
ORG	Companies, agencies, institutions, etc
GPE	Countries, cities, states
LOC	Non-GPE locations, mountains ranges, bodies of water
PRODUCT	Objects, vehicles, food, etc (Not services).
EVENT	Named hurricanes, battles wars, sports events, etc.
WORK OF ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates periods.
TIME	Time smaller than a day.
PERCENT	Percentage including ”%”
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	”first”, ”second”, etc
CARDINAL	Numerals that do not fall under another type.

References

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. *Reading wikipedia to answer opendomain questions.* In In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL).

[2] Diego Mollá, Menno van Zaanen, Daniel Smith. 2016. *Named Entity Recognition for Question Answering.* Centre for Language Technology, Macquarie University.

[3] Ana Cristina Mendes, Lu ́ısa Coheur, and Paula Vaz Lobo. 2010. *Named entity recognition in questions: Towards a golden collection.* In Proceedings of the International Conference on Language Resources and Evaluation.

[4] Elisa Noguera, Antonio Toral, Fernando Llopis, and Rafael Mu˜noz. 2005. *Reducing question answering*

input data using named entity recognition. In Proceedings of the 8th International Conference on Text, Speech Dialogue, pages 428–434.

[5] Robin Jia and Percy Liang. 2017. *Adversarial examples for evaluating reading comprehension systems.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. *Squad:100,000+ questions for machine comprehension of text.* In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP).