Detecting Character References in Literary Novels using a Two Stage Contextual Deep Learning approach

Krug, Markus
markus.krug@informatik.uni-wuerzburg.de
Chair of Applied Computer Science and Artificial Intelligence, Universität Würzburg, Deutschland

Kempf, Sebastian
sebastian.kempf@informatik.uni-wuerzburg.de
Chair of Applied Computer Science and Artificial Intelligence, Universität Würzburg, Deutschland

David, Schmidt
david.schmidt@informatik.uni-wuerzburg.de
Chair of Applied Computer Science and Artificial Intelligence, Universität Würzburg, Deutschland

Lukas, Weimer
lukas.weimer@uni-wuerzburg.de
Chair of Literary Computing, Universität Würzburg, Deutschland

Frank, Puppe
frank.puppe@informatik.uni-wuerzburg.de
Chair of Applied Computer Science and Artificial Intelligence, Universität Würzburg, Deutschland

Motivation

In recent years analyzing constellations of fictional entities in literary fiction has seen a lot of interest (Elson et al. 2010, Agarwal et al. 2012). Those constellations are often visualized as networks of entities apparent in the document. Even though the pipelines used for preprocessing vary drastically they all share some common steps. In order to draw a network, one needs to define nodes and edges. An obvious choice for the nodes are the fictional entities. Those entities appear either as pronouns, nominal references or names. The detection of those character references (CR) was used for a multitude of different applications in automatic digital humanities processing, most notably genre detection (Hettinger et al. 2015) and coreference resolution (e.g. Lee et al. 2013).

Related Work

This section only mentions the most dominant work in terms of Named Entity Recognition techniques as well as those with comparable results in terms of domain. The standard approach is to segment the input document into sentences and apply a sequence classifier on these sentences. The most robust classifier applied to this task was a Conditional Random Field (CRF) (Lafferty et al. 2001). Until the success of deep learning approaches, the classifier published by Stanford (Finkel et al. 2005) and its adaptation to German (Faruqui et al. 2010) were the dominant approaches. Lately the combination of a Bi-LSTM with word-embeddings (e.g. Mikolov et al. 2013) which automatically derived features for a CRF classifier in a deep learning approach surpassed manually created features (Huang et al. 2015). Most recently, Riedl and Padó (2018) included pretraining into the Bi-LSTM-CRF architecture and achieved state of the art results. A maximum entropy classifier with cluster features derived by word2vec and manually crafted features yielded an F1-score of 90% (Jannidis et al. 2017), serving as the only comparable work for German literary data.

However, all of these approaches classify each sentence and every token inside on their own so that subsequent sentences do not benefit from previously detected names. This does not only overcomplicate the detection, it can also introduce inconsistent results (e.g. “Effi” is detected as a name in sentence 10 and 27, but was not detected in sentence 25). Introducing no dependencies between each individual reference seems a wasted opportunity, especially in novels, because the same character reappears multiple times. This paper experimented with network architectures to leverage this shortcoming. The idea of this approach is not new and can even be dated back to the Brills Tagger (Brill 1992) - the classification by two separate classifiers each with its individual perspective on the problem.

Data

The corpus DROC (Krug et al. 2018) provides the data used in this paper. It contains about 393.000 tokens from 90 different samples taken from German novels. Each sample comprises at least one chapter. In total, this corpus contains about 53.000 manually annotated character references. 100-dimensional Word2Vec word embeddings trained on 1700 novels of project Gutenberg[1] were used as secondary input.

Methods

The method used in this paper follows the intuition that, especially in literary fiction, entities appear many times throughout the text. Because each document introduces its own fictional world, each word (meaning the set of all appearances of a token with the same string) has a

dominant meaning. However, not every instance of a word can be easily detected. While some might be surrounded by verbs of communication (“sagen”, “antworten”, ...), others might only be surrounded by stop words, which are not beneficial for classification. Therefore, this work introduces two passes through the text. The first pass tries to assign the dominant meaning to a word and is assumed to produce a high recall but a mediocre precision. The purpose of the second pass is to disambiguate individual instances which have been classified as a character reference but could have multiple senses. Furthermore, while the first pass might detect “Effi” and “Briest” as references, there is no information about whether the string “Effi Briest” is a single reference or two distinct references. This is solved in the second pass, which is trained for a sequential prediction and is supposed to detect the exact span of a reference.

The architectures of both neural networks can be depicted as follows:

An instance fed into the first network consists of a list of tuples, each comprising the span of the token, encoded by a word embedding, as well as a left context and a right context. Our previous work determined a context of the previous two and the next two tokens (also encoded by a pre-trained word embedding) as best performing for the determination of character references. The last input vector was derived by a Bi-LSTM character encoding of the target word. The tuples were arranged in order of appearance in the original text and encoded by a Bi-LSTM, feeding an additional tupel at each time step. The Bi-LSTM subsequently generates a condensed representation of those tuples into a vector of 256 units. The intuition is that this vector contains the most informative parts of all contexts for a given target word. The network is trained using log-loss and predicts whether the target word is a reference or not.

The network was trained for 15 epoches on 58 documents (longer training did not necessarily result in a better classification accuracy) and applied to a separate set spanning 14 documents. This second set is then used to train the second network, using Bi-LSTM character embeddings with a subsequent Bi-LSTM. However, the network is only applied to tokens that had been classified as a character reference in the first pass. This follows the intuition that it can now be decided if the current instance is of a different semantic category, which can be detected by analyzing its context. The input of this network is the snippet around the target word with a context size of two. The second task of this network, detecting the exact bounds of a reference, is done by predicting labels in an I-O-B setting. It is noteworthy that words that were not detected in the first pass can not be recovered. The second network is trained with 25 epochs and finally tested on 18 test documents.

Evaluation

We compared the architecture described in Section 4 (denoted 2-stage) with the state of the art architecture (Bi-LSTM-CRF) similar to Riedl and Padó (2018). A Bi-LSTM-CRF using character embeddings in Tensorflow[2] was implemented and applied to the data of DROC. At the current stage our implementation does not make use of pretraining. We used the 90 documents and split them into five folds each comprising 18 different test documents. The remaining 72 documents are used for training. The results are shown in Table 1.

System	Token			Entity		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
2-stage	89.6	74.9	80.7	89.2	69.5	78.1
Bi-LSTM-CRF	91.8	91.6	91.7	90.9	90.8	90.8

Table 1: Results of the two systems applied to DROC. The numbers were derived in a 5-fold scenario and are noted in %. Evaluation is done on token and on entity level using Precision, Recall and F1.

Even though the 2-stage approach seems intuitive at first, it can not compete with the results obtained by the state of the art architecture. Surprisingly, the architecture failed to provide a high recall (this is already apparent after the first pass, where the recall is similar to the state of the art system, however no exact borders can be predicted). A possible explanation for this result is the high amount of about 50% of tokens with only a single appearance in the text. Since only two context tokens to the left and the right are used, the architecture has a shortcoming compared to the Bi-LSTM-CRF, which encodes the entire sentence. The architecture does especially fail to recognize references that contain the token “von” (such as “Baron von Instetten)”. While being competitive in terms of the precision, further work has to be done to increase the recall for this approach.

Conclusion

This paper presents a 2-stage contextual approach to detect character references using deep learning. The results show that while the precision yields competitive results, the recall is still much lower. Possible approaches for this shortcoming might be changing the loss function - currently a false negative and a false positive yield the same penalty - and combining both models. The state of the art model can then be used for words that only appear a single time in the text and the 2-stage approach for words appearing more than once. This could retain the high quality while still generating a consistent labeling by making use of the dependencies between individual appearances of a word.

Fußnoten

1. https://www.gutenberg.org/
2. https://www.tensorflow.org/

Bibliographie

Agarwal, A., Corvalan, A., Jensen, J., & Rambow, O. (2012): *Social network analysis of alice in wonderland. In Proceedings of the NAACL-HLT 2012 Workshop on computational linguistics for literature* (pp. 88-96).

Brill, E. (1992, March): *A simple rule-based part of speech tagger*. In Proceedings of the third conference on Applied natural language processing (pp. 152-155). Association for Computational Linguistics.

Elson, D. K., Dames, N., & McKeown, K. R. (2010, July): *Extracting social networks from literary fiction*. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 138-147). Association for Computational Linguistics.

Faruqui, M., Padó, S., & Sprachverarbeitung, M. (2010, September): *Training and Evaluating a German Named Entity Recognizer with Semantic Generalization*. In KONVENS (pp. 129-133).

Finkel, J. R., Grenager, T., & Manning, C. (2005, June): *Incorporating non-local information into information extraction systems by gibbs sampling*. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 363-370). Association for Computational Linguistics.

Huang, Z., Xu, W., & Yu, K. (2015): *Bidirectional LSTM-CRF models for sequence tagging.* arXiv preprint arXiv:1508.01991.

Hettinger, L., Becker, M., Reger, I., Jannidis, F., & Hotho, A. (2015, September): *Genre classification on German novels*. In Database and Expert Systems Applications (DEXA), 2015 26th International Workshop on (pp. 249-253). IEEE.

Jannidis, F., Reger, I., Weimer, L., Krug, M., & Puppe, F. (2017): *Automatische Erkennung von Figuren in deutschsprachigen Romanen.*

Krug Markus, Puppe Frank, Reger Isabella, Weimer Lukas, Macharowsky Luisa, Feldhaus Stephan, Jannidis Fotis (2018, April): *Description of a Corpus of Character References in German Novels - DROC [Deutsches ROman Corpus]*. DARIAH-DE Working Papers Nr. 27. Göttingen: DARIAH-DE. URN: urn:nbn:de:gbv:7-dariah-2018-2-9

Lafferty, J., McCallum, A., & Pereira, F. C. (2001): *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.*

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., & Jurafsky, D. (2013): *Deterministic coreference resolution based on entity-centric, precision-ranked rules*. Computational Linguistics, 39(4), 885-916.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013): *Distributed representations of words and phrases and their compositionality*. In Advances in neural information processing systems (pp. 3111-3119).

Riedl, M., & Padó, S. (2018): *A Named Entity Recognition Shootout for German*. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 120-125).