Design Challenges in Wikipedia-style Spotlight Annotation

Kunyuan Pang¹, Jintao Tang¹, Xinyi Li², Xiao Li³, Shasha Li¹ and Ting Wang¹*

¹ College of Computer Science and Technology
² Science and Technology on Information Systems Engineering Laboratory
³ Information Center National University of Defense Technology, Changsha, Hunan, 410073, China

*Corresponding author’s e-mail: tingwang@nudt.edu.cn

Abstract. Wikipedia has set up a manual of style to provide the Wikipedian with guidelines to create links as spotlights in articles to help readers' understanding. Wikification is coined following Wikipedia’s style guide as a task of automatically detecting mentions and linking to entities in plain-text articles. However, existing methods rely heavily on the mention-entity dictionary and focus on in-map entity precision. Some popular methods are limited to Named Entities. The comparison on the outcome of spotlight annotation is insufficient. We investigate the design challenges for constructing automatic annotating in Wikipedia-style systems. A dataset is built simulating real-world spotlight annotation problems in both the preparing and application phases. Popular methods are compared along with a straightforward framework we propose called WikiER. Experiments show that the performance of currently available systems varies a lot. It is important to train on the task-specific data. The best performance is achieved by WikiER.

1. Introduction

Wikipedia has become the most influential online encyclopedia in recent years. The linking nature of Wikipedia makes it easier for readers to understand the article and explore relevant topics. It helps readers to notice what entities are in the current article as well as clarify misleading names such as geographical locations and technical terms. Editors create links by choosing pieces of text (called mention) in a Wikipedia article and linking each of them to its corresponding Wikipedia article (called entity). Wikipedia has set up a manual of style to provide guidelines on how to create links in articles. The main idea is to choose only the most helpful and non-repetitive mention-entity links for readers, as is shown in Figure 1. We call the selected mentions spotlights.

Thanks to the detailed guidelines and the collaboration of Wikipedian, many high-quality Wikipedia articles have been considered as typical examples of how to spotlight and link to a knowledge base (KB). Since manually creating links is tremendously laborious and needs expert experience, Wikification has been proposed as a research task[1-3] as automatically identifying concept mentions appearing in a text document and link it to an entity in Wikipedia. Unfortunately, despite abundant papers and datasets on the topic, there is not enough understanding about how well these methods adhere to the Wikipedia guidelines regarding helping readers.

We argue that this confusion is caused by simplifications in different research in three major aspects. First, the problem is simplified to annotate only named entities. The most studied variant is Named Entity Linking, which targets only a few types[4]. Secondly, the problem is simplified and studied in a closed scenario with impractical additional data. For example, [5] requires a mention-entity map and
ignores mentions without a valid KB entity. Finally, the selection is often ignored. Many tasks are defined as detecting all mentions in the text.

To address these problems, we build a real-world dataset focusing on the outcome of spotlight annotation. The test set cases are real text sampled from Wikipedia articles of the best quality (aka featured articles). To simulate the real-world practice, we fix the training set as the rest featured articles. This dataset simulates the scenario where given the annotation guideline (Wikipedia guideline) and adequate human-annotated samples (train-set), how to work on new-come articles (test-set). We propose an end-to-end spotlight annotation system called Wikipedia Entity Recognizer (WikiER). We compare our model with existing popular systems with the outcome in spotlight annotation.

Our contributions are:
- We analyze the differences between the existing popular systems and the real-world engineering problem of spotlight annotation in task definition and actual performance. (Section 2 & 5)
- We construct a dataset simulating the real-world engineering problem of spotlight annotation. The test set focus on the actual performance. The train set ensures the same conditions for systems. (Section 4)
- We present an end-to-end spotlight annotation system, WikiER. We will make the dataset and the code open-source and publicly available. (Section 3)

2. Mention Selection in Wikipedia-style Spotlight

In this section, we discuss the related work and popularly adopted systems and compare their prerequisite, task settings with the actual problem of spotlight annotation. Table 1 summarizes statistics of related tasks (open challenge tasks and task definition of specific systems).

2.1. Mention and Entity Types

In real-world Wikipedia, people choose any phrase that can link to another Wikipedia article. A well-studied mention extraction task is called Named Entity Recognition (NER), which extracts only direct names of a named entity of a few appointed types. In AIDA[6], the types are person(PER), organization(ORG) and location(LOC). The entity type of traditional Entity Discovery and Linking (EDL) tasks is similar. Fine-grained Named Entity Recognition expands to more entity types (up to 16,000) [7].

Table 1. Statistics of related tasks and systems.

| Task          | Typical Systems | Entity Types | Entity Scope | Mention Selection | Additional resource for mention extraction |
|---------------|-----------------|--------------|--------------|-------------------|--------------------------------------------|
| NER/EDL       | Flair           | PER, ORG, GPE| All          | No                | No need                                    |
| Fine-grained NER | GAIA           | 20-16,000 predefined fine-grained types | All | No | No need |
Limiting the type to named entities brings convenience to constructing datasets, which also benefits both testing and training of the model. But in the real world, it leads to ignoring necessary knowledge in the text.

2.2. KB and Additional Required resource

Many tasks and systems are designed surrounding a fixed KB and probabilistic mention-entity map, [3-6,8,17,19] so systems can focus on ranking candidates. The problem is simplified because it is clearer which spans are possible mentions and the information of entities is abundant in KB.

Other tasks assume to have gold annotation from another task as part of their pipeline. In salient entity tasks[8], they take given mentions and entities and focus on entity scoring. In so-called zero-shot entity linking[9,10], they take given mentions and focus on linking.

In a real-world application, it is hard to get such gold resources. The missing terms in KB, mention-entity map, and errors in upstream tools accumulate.

2.3. Mention selection

Mention selection is the core requirement of Wikipedia’s guidelines. Wikipedia Miner[3] try to model the selection after entity linking with entity selection scores. Other tasks are coined detecting all mentions. This will cause false negatives in the real-world application. Besides Wikipedia’s guide of choosing helpful mentions, salient entity selection is another criterion of selection. A salient entity is defined as entities most relevant to the article[8, 11]. We will check how this different salience definition works for our spotlight annotation task.

3. Methodology of WikiER

In this section, we present WikiER, a straight-forward Spotlight mention detector. As input, the model takes plain text. We adopt a BERT-CRF model to solve mention selection as a sequence labeling problem. BERT[12] takes tokenized word sequence as input and generates context embedding for each word. We use this context embedding to conduct sequence labeling with BIO labeling schema[13] using conditional random field (CRF) [14] following [15].

Specifically, for a sentence with $n_w$ words $[w_1, w_2, ..., w_{n_w}]$, we predict the BIO labels for the words denoted as $[y_1, y_2, ..., y_{n_w}]$ or $y$ for short.

First, we define the potential function in Equation 1, $W_{y_{l-1}}^{y_{l}}$ and $b^{y_{l}}_{y_{l}}$ are trainable parameters. The probability to predict the output label sequence $y$ with BERT embedding $H$ is calculated in Equation 2, and the loss of predicting the whole sequence is calculated in Equation 3. The best prediction of $y$ is the sequence that maximizes $p(y|H)$. We search for the sequence by the Viterbi algorithm [16].

$$
\psi_l(y_{l-1}, y_l, h_l) = \exp (W_{y_{l-1}}^{y_{l}} h_l + b^{y_{l}}_{y_{l}})
$$

(1)

$$
p(y|H) = \frac{\prod_{i=1}^{n_w} \psi_l(y_{l-1}, y_l, h_l)}{\Sigma_{y' \in \mathcal{Y}} \prod_{i=1}^{n_w} \psi_l(y'_{l-1}, y'_l, h_l)}
$$

(2)

$$
L = -\log p(y|H)
$$

(3)
4. Experiments

4.1. Dataset

We construct the WikiFA-ms dataset for overall performance comparison and the WikiNM dataset for new mention coverage evaluation. Specifically, the WikiFA-ms test set and develop set are real texts and annotations extracted from 100/200 randomly sampled featured articles from the Wikipedia dump in 2019. The WikiFA-ms train-set is built with real texts and annotations extracted from the rest featured articles.

In WikiFA-ms, we train the models on the same train set whenever possible and evaluate with precision, recall, and F1-measure on the mention phrases of each article. This setting is the most similar to when a practitioner is given an annotation guideline (Wikipedia guideline), prepares human-annotated samples, and tries to automatically produce annotation for articles in a working environment.

The WikiNM dataset is a subset of WikiFA-ms to test the model’s ability on new mentions. We extract cases of WikiFA-ms whose mentions are in the map extracted from the 2019 Wikipedia dump but not in the 2015 map and are easier to be out of the map mentions. We evaluate the recall rate of mentions on WikiNM.

4.2. Systems to compare

- Wikipedia-miner (WM) [3] is the traditional Wikification model depending on a mention-entity map extracted from the whole 2015 Wikipedia. We train its entity selection classifier with the development set as they described. It is worth noticing that WM still works in a closed scenario in our experiment, as the crucial information such as the mention-entity map and entity relation highly overlap the test-set.
- TAGME [17] and WAT [19] are popular Wikification online APIs that return mentions with annotation confidence.
- SWAT [8] is an online API that re-scores WAT results with their definition of saliency.
- Flair [18] leads most NER benchmarks. The default NER results of Flair are reported in our experiments (Flair-ori). We also train Flair with WikiFA-ms train set to test whether NER methods solve mention selection (Flair-wiki).
- GAIA [7] is the newest fine-grained NER system. They provide the model trained on Wikipedia as silver-standard and finetune with fine-grained named entity annotation.
- WikiER is the system we proposed in Section 3 and trained with only WikiFA-ms train set.

4.3. Analysis

Table 2. Mention Selection Performance.

| System       | WikiFA-ms | WikiNM | Preparing time cost | Local deployment |
|--------------|-----------|--------|---------------------|-----------------|
|              | Precision | Recall | F1                  | Recall          |                  |
| WM           | 0.888     | 0.809  | 0.847               | 0.000           | ~2 days          |
| TAGME        | 0.684     | 0.701  | 0.692               | 0.336           | 0                |
| WAT          | 0.328     | 0.336  | 0.332               | 0.040           | 0                |
| SWAT         | 0.155     | 0.541  | 0.241               | 0.249           | 0                |
| GAIA         | 0.229     | 0.430  | 0.299               | 0.325           | 0+Download time |
| Flair-ori    | 0.593     | 0.478  | 0.530               | 0.270           | 0+Download time |
| Flair-wiki   | 0.760     | 0.634  | 0.691               | 0.398           | Hours            |
| WikiER(ours) | 0.808     | 0.729  | 0.767               | 0.483           | ~1 day           |
Table 2 shows the performance and usability of different systems. In overall performance, our WikiER model significantly outperforms online map-dependent Wikification methods (TAGME and WAT), the default NER method (GAIA and Flair-ori), and the trained NER model (Flair-wiki). It is close to the close scenario WM.

For new mentions, WM fails to detect any new mention for complete dependence on the mention-entity map. WAT and SWAT return the same mention set with different scores. WAT returns annotation confidence while SWAT returns its entity saliency. The performance drop reveals that the definition between Wikipedia guidelines and entity saliency is not quite relevant. Comparing GAIA and Flair-ori shows spotlight annotation is not simply adding named entity of more types.

The usability of systems is also reported as it is important for practical engineering. For preparation time, online APIs are instant to use. GAIA and the original Flair take some downloading time depending on the network speed. Flair and WikiER need model training, which takes less than 24 hours. WM takes the most time as it extracts the mention-entity map from Wikipedia dump. Practitioners might need to deploy the model on an appointed server for safety and convenience. It is impossible for APIs. GAIA, Flair, and WikiER require the server to be installed with GPUs to achieve reasonable speed.

5. Conclusion
We examine popular Wikification systems in computer engineering and discuss their difference with the real-world spotlight annotation problem. We construct a dataset simulating the real-world engineering problem. Evaluation on the dataset shows existing APIs and models does not solve the real-world problem as expected; training on the prepared data significantly improves the performance and our proposed model WikiER achieves the best performance after all.

Our future work includes two directions. (1) We plan to study the positions of mentions based on precise mention selection. (2) We plan to combine our method with resources like mention-entity maps, to take advantage of the enriched features on in-map mentions, too.

References
[1] Mihalcea, R., Csomai, A., (2007). Wikify!: linking documents to encyclopedic knowledge, In CIKM’07. pp. 165-178.
[2] Ratinov, L., Roth, D., Downey, D., Anderson, M., (2011). Local and global algorithms for disambiguation to Wikipedia, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, pp. 1375-1384.
[3] Milne, D., Witten, I.H., 2008. Learning to link with Wikipedia, Proceedings of the 17th ACM conference on Information and knowledge management. ACM, pp. 509-518.
[4] Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J., 2010. Overview of the TAC 2010 knowledge base population track, Third Text Analysis Conference (TAC 2010), p. 3.3.
[5] Kolitsas, N., Ganea, O.-E., Hofmann, T., 2018. End-to-End Neural Entity Linking, Proceedings of the 22nd Conference on Computational Natural Language Learning, pp. 519-529.
[6] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenuh, H., Pinkal, M., Spaniol, M., . . . Weikum, G. (2011). Robust disambiguation of named entities in text. Paper presented at the Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing.
[7] Li, M., Zareian, A., Lin, Y., Pan, X., Whitehead, S., Chen, B., . . . Voss, C. (2020). Gaia: A fine-grained multimedia knowledge extraction system. Paper presented at the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations.
[8] Ponza, M., Ferragina, P., & Piccinno, F. (2018). SWAT: A System for Detecting Salient Wikipedia Entities in Texts. arXiv preprint arXiv:1804.03580.
[9] Wu, L., Petroni, F., Josifoski, M., Riedel, S., & Zettlemoyer, L. (2020). Scalable Zero-shot Entity Linking with Dense Entity Retrieval. Paper presented at the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).
[10] Logeswaran, L., Chang, M.-W., Lee, K., Toutanova, K., Devlin, J., & Lee, H. (2019). Zero-Shot Entity Linking by Reading Entity Descriptions. Paper presented at the Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

[11] Dunietz, J., & Gillick, D. (2014). A new entity salience task with millions of training examples. Paper presented at the Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers. pp. 205-209.

[12] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) pp. 4171-4186.

[13] Ramshaw, L., & Marcus, M. (1995). Text Chunking using Transformation-Based Learning. In Third Workshop on Very Large Corpora.

[14] Lafferty, J., McCallum, A., & Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.

[15] Ma, X., & Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. Paper presented at the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

[16] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE transactions on Information Theory, 13(2), 260-269.

[17] Ferragina, P., & Scaiella, U. (2010). Tagme: on-the-fly annotation of short text fragments (by Wikipedia entities). Paper presented at the Proceedings of the 19th ACM international conference on Information and knowledge management.

[18] Akbik, A., Bergmann, T., & Vollgraf, R. (2019, June). Pooled contextualized embeddings for named entity recognition. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 724-728).

[19] Piccinno, F., & Ferragina, P. (2014). From TagME to WAT: a new entity annotator. Paper presented at the Proceedings of the first international workshop on Entity recognition & disambiguation.