SU-NLP at SemEval-2022 Task 11: Complex Named Entity Recognition with Entity Linking

Buse Çarık*, **Fatih Beyhan***, **Reyyan Yeniterzi**
Sabancı University, İstanbul, Turkey
{busecarik,fatihbeyhan,reyyan}@sabanciuniv.edu

Abstract

This paper describes the system proposed by Sabancı University Natural Language Processing Group in the SemEval-2022 MultiCoNER task. We developed an unsupervised entity linking pipeline that detects potential entity mentions with the help of Wikipedia and also uses the corresponding Wikipedia context to help the classifier in finding the named entity type of that mention. The proposed pipeline significantly improved the performance, especially for complex entities in low-context settings.

1 Introduction

Named Entity Recognition (NER) is a widely studied task of Natural Language Processing. Recent leading architectures achieved impressive performances, especially in formal contexts (like news articles, etc.) and for commonly worked named entity types (like Person, Organization, etc.). In *Multilingual Complex Named Entity Recognition* task (Malmasi et al., 2022b), organizers introduced a dataset which is a large collection of short texts that includes complex entities. As they stated, complex entities can be noun phrases, gerunds, or full clauses (Malmasi et al., 2022a). The participants were challenged to develop NER systems for identifying these complex entities in short and low-context settings.

Recent contextual neural architectures like transformers work quite well on a variety of NLP tasks and datasets. However, they are strongly dependent on the context of the given text. In case of lack of context, these models alone may not gather enough evidence to make a correct prediction. When that is the case, getting help from knowledge bases can be useful. The dataset provided with this task has similar low-context characteristics together with some complex named entity types to be resolved. Therefore, in this work, we investigated whether performing entity linking to provide additional context for a transformer model can be useful for identifying complex named entities. We initially used an unsupervised entity linking approach to identify the possible entity mentions and their corresponding Wikipedia pages. Later on, we used the additional context retrieved from Wikipedia to predict the types of the identified entities. Performing entity linking prior to the classifier seems to be beneficial in the case of complex entities in short and low-context settings. Our code is publicly available*.

2 Related Work

NER is an important and also well-studied task in NLP. Similar to other NLP tasks, recent neural architectures provided significant improvements in NER as well. However, since these models are heavily dependent on context information, applying them to short, noisy texts such as very short social media posts or web queries results in significant performance degradation (Meng et al., 2021).

To overcome the lack of context in these short texts, as well as the code-mixed terms, it has been prominent to use external knowledge in neural approaches. Recent studies (Meng et al., 2021; Fetahu et al., 2021) combining multilingual transformer models with gazetteers have shown remarkable improvement in performance. Another study (Yamada et al., 2015) utilized the Entity Linking task to address the issues of Twitter NER by detecting entity mentions from Wikipedia.

In this study, we also focused on external information due to the lack of context. However, instead of using a static auxiliary like gazetteers, we tried to detect entities dynamically from the continuously extending Wikipedia by using a search engine architecture. We integrated the Wikipedia content into the input representation in order to improve the performance of entity type classification.

*These authors contributed equally to this work.

*https://github.com/SU-NLP/SU-NLP-at-SemEval-2022-Task-11-Complex-Named-Entity-Recognition-with-Entity-Linking

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1648 - 1653

3 Data

Malmasi et al. (2022a) introduced a new multilingual NER dataset consisting of 6 named entity categories (Person, Location, Group, Corporation, Product, and Creative Work) annotated in short texts for 11 languages. In this paper, even though we propose a language-independent approach, due to the limited resources we only focused on the Turkish part of the dataset. Statistics from this part are shown in Table 1. As seen from Table 1, even though the number of samples is much higher, the length of the samples is significantly lower in the test set compared to train and validation splits. This short length in context may cause state-of-the-art contextual models to perform worse than expected.

	# Samples	Avg. # Tokens
Train	15,300	14.27
Validation	800	14.27
Test	136,935	5.28

Table 1: Distribution of samples and average number of tokens per sample for Turkish part

An issue we found with the dataset is that in some examples, a phrase is labeled as *OTHER* although there are similar phrases that are labeled as a named entity. Consider the following example: *"Nokia 1011, 1994'e kadar, Nokia 2010 ve Nokia 2110'un halefler olarak tanıtılmasıyla üretime devam etti." English: "Nokia 1011 continued in production until 1994, with the introduction of the Nokia 2010 and Nokia 2110 as successors."*

The text includes three different cellphone models produced by Nokia. Two of these models (the blue ones) were annotated as *PRODUCT*; however, *Nokia 1011* was mislabeled as *OTHER*. A more major issue is when the same named entity occurs more than once in a sentence, but their annotations are different. An example is illustrated below: *"Whitney Houston yine bu sene içinde New Jack Swing'e adını yazdırdı, I'm Your Baby Tonight adlı şarkı Whitney Houston'a başarı üstüne başarı getirmiştir." English: "Whitney Houston made her name on New Jack Swing again this year, the song I'm Your Baby Tonight brought Whitney Houston success after success."*

In the example, Whitney Houston in blue was labeled as *PERSON*; but, the red Whitney Houston was labeled as *OTHER*. These types of wrong annotations cause ambiguities and learning problems in supervised approaches.

4 Methodology

In order to overcome the challenges described in previous sections, we propose an unsupervised entity linking pipeline followed by a classifier as our proposed NER architecture. The entity linking pipeline is designed to detect potential entity mentions and retrieve corresponding candidate documents for each mention. Later, these detected mentions and the linked content is used together in a classifier to detect the entity type of the mention. A pre-trained BERT model was also fine-tuned in order to create a strong baseline for comparison.

4.1 BERT

The BERT (Devlin et al., 2018) model was adopted as a baseline approach due to its outstanding performance in many NLP tasks. We fine-tuned the BERTurk (Schweter, 2020) model with a feed-forward layer on top to classify the named entities. The base model with 12 encoder layers and 768 hidden units was used. Since the data provided as part of the task was all lowercase, the uncased version of BERTurk was preferred.

4.2 NER with Entity Linking

BERT like transformer models heavily depend on the context. Therefore having a very short context (like a couple of words, not even a sentence) can be challenging for these models. Especially when the task is to identify complex named entities like creative work or product, it is much more challenging. In a very short context, even human beings may have a hard time resolving entities and their types.

In order to overcome this challenge, additional help from a gazetteer or a knowledge base will be very useful. Therefore in this paper, we propose using Wikipedia in order to both identify entities and also their types. Our proposed architecture which is depicted in Figure 1 consists of three steps. The first two steps work towards identifying possible entities and linking them to their corresponding Wikipedia articles. In this unsupervised entity linking approach as the first step, the original input text is used to retrieve relevant Wikipedia pages which are possible candidate pages for entities. In the second step, content within the retrieved documents (like their titles) is used to identify the entity mentions in the original text. Finally, in the third step, each identified mention and its linked Wikipedia article are used together to predict the type of the

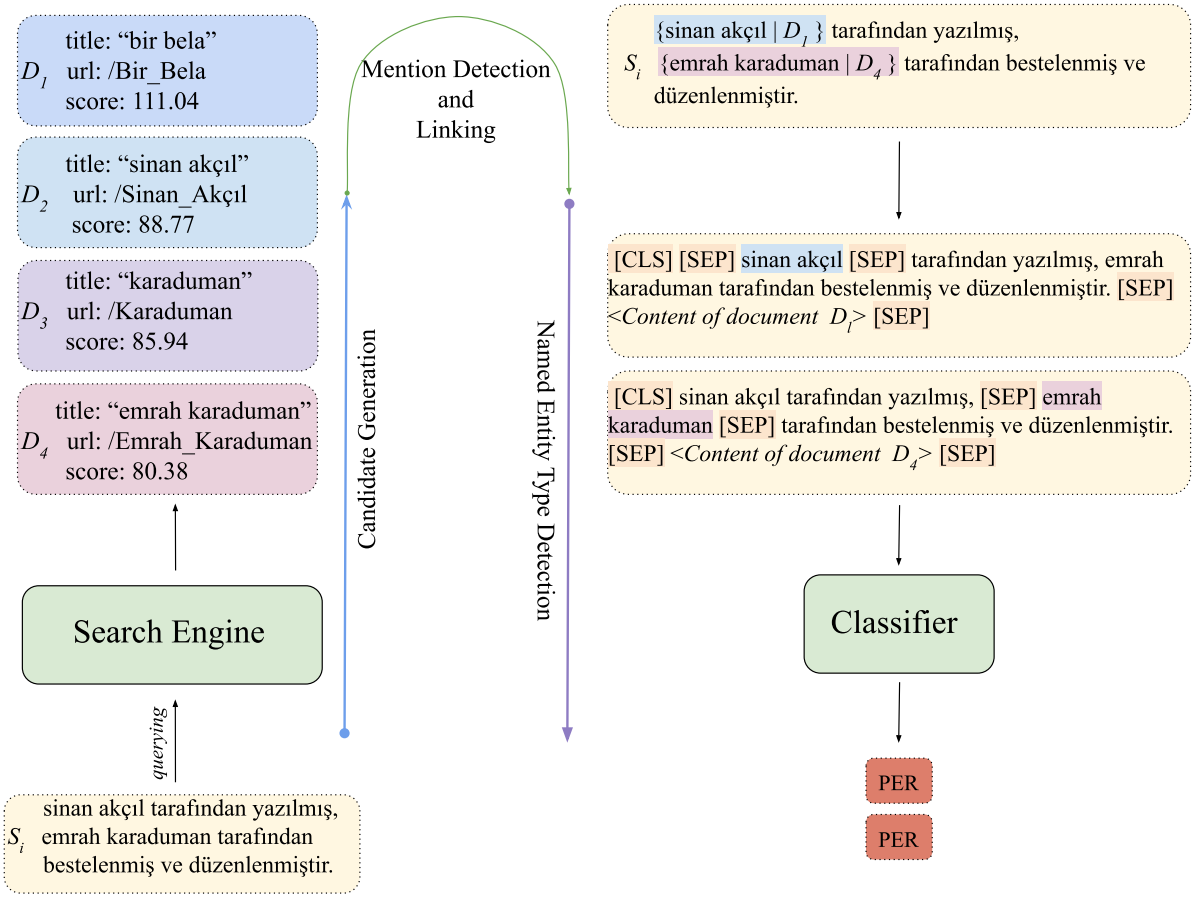

Figure 1: Overall pipeline of the system which is showing the path of sentence S_i.

entity.

4.2.1 Candidate Generation

ElasticSearch[*] was used as the underlying search engine architecture. The most recent Wikipedia dump was used to create an index. For each document D_i, the *title*, *referred_by* and *interwikies* were indexed to create the following four fields:

- *title*: This is the title of the document D_i.
- *referred_by*: This is a set of text spans which are parsed from other Wikipedia pages where D_i is being referred with a interwiki link. This set includes all of these text spans sorted from longest to shortest. The title is added to the set as well.
- *interwikies*: This is the list of interwikies[*] in the document D_i.
- *all_text*: Concatenation of all fields (title, text content, interwikies, referred_by, and categories[*]) of the document D_i.

Each sample was queried and the most relevant 200 articles were retrieved. Later on for each query, documents retrieved with different field searches were pooled together. The reason we are pooling these different field results is to make sure we do not miss a possibly relevant article that could have been linked.

For the train and validation tests, around 199 documents were retrieved for each field. After pooling, we ended up with a pool of size 578 documents on average. Since test queries are shorter, the average number of documents returned got as low as 161 with the title field and 465 over the pool. Short queries also affected the empty ones, in other words, no document retrieved samples. After pooling there was not such a case for train and validation, however even pooling could not help the 169

[*]https://www.elastic.co/

[*]Interwikies are the types of hyperlinks which is linking a text span from a Wikipedia page to another Wikipedia page. For instance, Wikipedia pages of Earth, Mercury and Venus are in the interwikies section of the Sun’s Wikipedia page, since they are mentioned within its text.

[*]The categories section of a Wikipedia page includes a list of topics which is related to the Wikipedia page.

Field	Train	Validation
title	96.52	97.00
interwikies	57.16	61.26
referred_by	96.46	96.11
all_text	86.50	86.68
After pooling	98.89	98.85

Table 2: Recall Scores over Detected Mentions

queries over 136,935 samples, and no Wikipedia articles were retrieved for those.

4.2.2 Mention Detection and Linking

After generating a set of candidate documents for each input text, the next step is to map these documents to the possible entity mentions in the input. For each input, our proposed algorithm iterated over the retrieved and pooled documents $\{D\}$, and cross-checked each item in the *referred_by* field of the document to see if there was an exact match to any phrase in the input text. In case there was an exact match, then that matched phrase was tagged as a possible entity and linked to the corresponding Wikipedia page. If there were multiple matches for a span of text, then the longest match was considered and shorter ones were ignored. Furthermore, if there was a tie in terms of length, then the relevance score of the Wikipedia document was used. That relevance score was calculated after pooling by doing a summation of the relevance scores from different field retrievals.

In order to see which fields in Wikipedia are more useful for mention detection, recall was calculated over detected entity mentions, without considering the type of entity. The results are summarized in Table 2. Based on the results *title* and *referred_by* are the most useful fields and pooling all retrievals is the best over both train and development sets.

4.2.3 Named Entity Type Detection

After linking the entities, the only thing remaining is to find their NER type. In order to do that we used the original context where the entity was mentioned in the input text and the linked Wikipedia article content. We combined these two as follows:

$$\texttt{[CLS]}\, ctx_l\, \texttt{[SEP]}\, M\, \texttt{[SEP]}\, ctx_r\, \texttt{[SEP]}\, WP\, \texttt{[SEP]}$$

where ctx_l and ctx_r are tokens for left and right context of mention, M is mention itself and WP is the first two paragraphs of a Wikipedia page. `[SEP]` tokens are used to tag the mention. The maximum length of the representation is set to 256. This representation is given to BERT for encoding.

We used two classifier heads following the BERT encoder for classifying the pairs of mentions M and Wikipedia contents WP. The proposed architectural design is presented in Figure 2. The first one is a binary classifier head which aims to solve an auxiliary task of whether the given candidate entity mention is a real entity or not with the help of input context tokens and Wikipedia content. The second classifier head tries to learn the type of the candidate named entity. The loss function for our model is the summation of losses from these two classifier heads. In case the binary classifier head returns O (means mention is not an entity), the second classifier head's decision is ignored, and the given candidate entity mention is directly predicted as O (which stands for Other, not an entity). This multi head model is referred to as *EL_MultiBERT*.

In order to test the effectiveness of multiple heads, we created another model called *EL_BERT*, which has a single classifier head only for identifying named entity types including the Other class.

In our pipeline, the Other class can be assigned to tokens in any of the three steps. Whenever a token is assigned the Other (O) label, then its label does not change during the remaining steps.

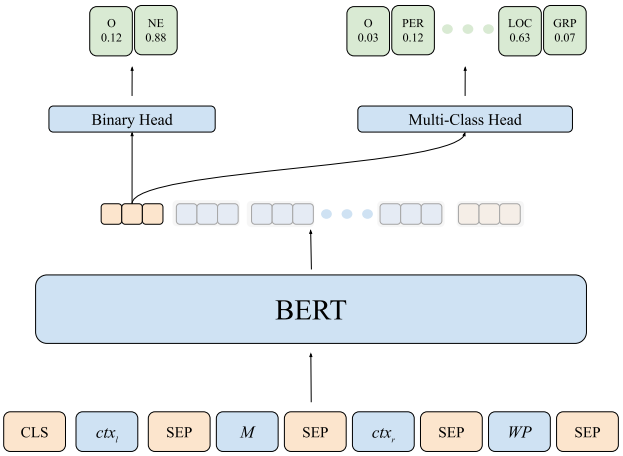

Figure 2: The architecture of the classifier. NE in the binary classifier head's output stands for *Named Entity* and O for *Other*.

5 Experimental Setup

Due to limited resources and time, we only experimented with the Turkish part of the dataset and therefore indexed 749,204 Turkish Wikipedia pages. During retrieval, *OR* operator was used as

Models	Validation			Test		
	Precision	Recall	F1	Precision	Recall	F1
BERT	85.84	85.01	**85.39**	58.02	58.39	57.82
EL_BERT	84.59	79.40	81.74	74.78	59.03	65.47
EL_MultiBERT	**86.18**	77.56	81.60	**80.69**	65.13	71.91
Ensemble$_{Token+EL}$	83.78	85.35	84.47	74.23	59.44	65.66
Ensemble$_{Token+EL_Multi}$	83.66	**85.51**	84.49	78.86	**66.43**	**72.02**

Table 3: Results of our proposed methods on development and test data

part of the query and the default BM25 was used for ranking. We fine-tuned the BERTurk model[*]. As the optimizer, we used AdamW (Loshchilov and Hutter, 2019) with a fixed learning rate in all our BERT models as 3 x 10^{-5}. We also applied dropout before the feed-forward layer with 0.3 probability. Our batch size was 32, with a maximum sequence length of 128 for the baseline BERT. We extended the maximum sequence length to 256 for EL_MultiBERT and reduced the batch size to 8. All experiments were conducted with seed 22.

6 Results

The official evaluation metric was chosen as macro-average F1-score by the organizers. In Table 3, we present our results with models described above. Our baseline BERT, the one fine-tuned directly for NER, worked quite well on the validation set but performed badly on the test set. This substantial difference in these two sets shows that contextual models like BERT do not perform well in low-context settings. On the other hand, although our proposed entity linking approaches fell slightly behind the baseline model in the development set, they performed significantly better than the baseline on test set. Among the two entity linking methods, the EL_MultiBERT model with two classification heads performed better. While determining the named entity type, training with this auxiliary task even on the same data seems to be useful.

Table 4 illustrates the performance of BERT and EL_MultiBERT models for each named entity class. As seen from the table, our proposed approach improves the performance, especially in complex entities like *Creative Work*, *Group* and *Product*. Two approaches returned similar results only for *Person* and *Location*. Since we do not have the gold standard labels, we cannot perform any detailed analysis of the reason.

[*]https://huggingface.co/dbmdz/bert-base-turkish-uncased

Entity Type	BERT	EL_MultiBERT
Person	70.72	72.43
Location	59.73	57.42
Group	51.49	76.51
Corporation	57.69	73.79
Product	63.13	76.93
Creative Work	44.14	74.40

Table 4: F1-Scores by NER type on Test Set

6.1 Ensemble Approach

Although the majority of the samples in the test set are short texts, there are some longer sentences as well. From the validation set experiments, we already know that when there is enough context, transformer models like BERT perform quite well. Hence, we decided to see if we can use our proposed architecture as a fall back mechanism when there is not enough context available, but otherwise use BERT. We applied an ensemble approach relying on BERT for samples with more than 11 tokens and EL_MultiBERT for shorter ones. So when sentence length is less than 11, we depended on our entity linking architecture and Wikipedia for retrieving and using additional useful context about the input. As shown in Table 3, the ensemble of BERT and EL_MultiBERT achieved 72.02% on the test set, and this system ranked as third in the Turkish track.

7 Conclusion

In this work, we proposed a language-independent method for the Complex NER task in low-context settings. Our results showed that utilizing the use of entity linking for NER provides a significant improvement over state-of-the-art transformer models when there is not enough context. Yet, since our resources are limited, we experimented with only the Turkish part. In future work, we will extend this approach in a multilingual setting.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. *Gazetteer Enhanced Named Entity Recognition for Code-Mixed Web Queries*, page 1677–1681. Association for Computing Machinery, New York, NY, USA.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. MultiCoNER: a Large-scale Multilingual dataset for Complex Named Entity Recognition.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512, Online. Association for Computational Linguistics.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. Enhancing named entity recognition in Twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 136–140, Beijing, China. Association for Computational Linguistics.