Approaching Nested Named Entity Recognition with Parallel LSTM-CRFs

Łukasz Borchmann, Andrzej Gretkowski, Filip Graliński (APPLICA.AI)

Abstract

We present the winning system of this year's PolEval nested named entity competition, as well as the justification of handling the particular problem with multiple models rather than relying on dedicated architectures. The description of working out the final solution (parallel LSTM-CRFs utilizing GloVe and Contextual Word Embeddings) is preceded with information regarding recent advances in flat and nested named entity recognition. Significantly, all the tested solutions were developed on the basis of open source implementations, particularly Flair framework, LM-LSTM-CRF, Layered-LSTM-CRF and Vowpal Wabbit.

Keywords

nested named entity, named entity recognition, LSTM-CRF, contextual word embeddings, Polish NER, GloVe embeddings

1. Introduction

Named entity recognition (or entity identification, entity chunking, entity extraction) is a task of locating and classifying spans of text associated with real-world objects, such as person names, organizations and locations, as well as with abstract temporal and numerical expressions (eg. dates).

1.1. Flat Named Entity Recognition

As Young et al. (2017) summarize, after decades of *machine learning approaches utilizing shallow models trained on high dimensional and sparse features*,[1] came time of neural networks based on dense vector representations. It is also the case for named entity recognition systems, where those relying on hand-crafted features and domain-specific resources can be outperformed with simple deep learning frameworks.[2]

Many modern and successful NER solutions follow Huang et al. (2015) and Lample et al. (2016) approaching task with bidirectional LSTM-CRF architecture, which proved to be a strong candidate for structured prediction problems.

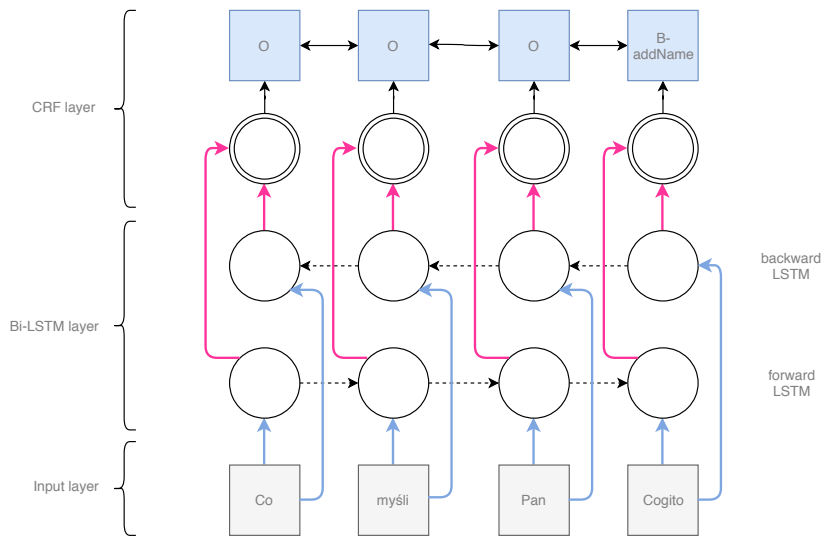

Figure 1: BiLSTM-CRF architecture (Huang et al. 2015, Lample et al. 2016).

Table 1 presents the results of the selected LSTM-CRF-based solutions in the CoNLL 2003 NER task. Liu et al. (2018) showed that LSTM-CRF architecture can be empowered by training a character-level language model at the same time, in addition to the sequence labeling model. Recent approaches by Peters et al. (2018) and Akbik et al. (2018) use embeddings obtained from internal states of deep language models pre-trained on a large text corpus. These are expected to capture context-dependent word semantics.

A common approach is to stack conceptually different embeddings, eg. by concatenating LM's embeddings with count-based approaches of obtaining vector representations

[1] Cf. eg. (Nadeau and Sekine 2007) for a review of pre-neural solutions.

[2] There are, however, also some attempts to incorporate domain-specific knowledge, eg. by injecting it into word embeddings (Celikyilmaz et al. 2015, Pandey et al. 2017).

Table 1: Results of selected LSTM-CRF-based solutions in the CoNLL 2003 NER task.

Method	**Span F_1**
Contextual string embeddings (Akbik et al. 2018)	93.09
Deep contextualized word representations (Peters et al. 2018)	92.22
Task-aware neural language model (Liu et al. 2018)	91.71
Classic LSTM-CRF (Lample et al. 2016)	90.94

for words, such as GloVe proposed by Pennington et al. (2014). According to the distributional hypothesis, *difference of meaning correlates with difference of distribution* (Harris 1954), that is words sharing context tent to share similar meanings, which is often perceived as theoretical justification of the former representations.

The current state-of-the-art was established by Akbik et al. (2018) with contextualized string embeddings stacked with GloVe embeddings for English and fastText embeddings for German language (Bojanowski et al. 2017).

1.2. Nested Entity Identification

The methods described above receive particular attention of researchers and are the basis of related nested named entity recognition systems, where it is expected that named entities can overlap and contain other named entities. Figure 2 presents an example of such coming from the National Corpus of Polish (Przepiórkowski et al. 2012), namely street name (here classified as *geogName*), consisting of a person name (*persName*), containing *forename* and *surname*.

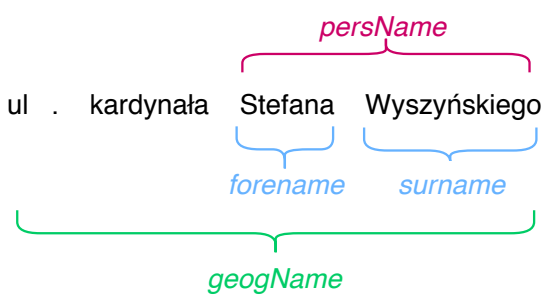

Figure 2: Example of nested named entity from the National Corpus of Polish (*ul. kardynała Stefana Wyszyńskiego* 'Cardinal Stefan Wyszyński Street').

These were proposed to be handled in multiple ways, whereas many of them rely on an old paradigm of handcrafted features, such as cascaded CRF model, constituency

parser with constituents for each named entity or mention hypergraph model (Katiyar and Cardie 2018). Recently however, the problem was successfully addressed with neural architectures, by dynamically stacking additional flat CRF layers in LSTM-CRF model (Ju et al. 2018) and learning the entity hypergraph structure (Katiyar and Cardie 2018).

1.3. PolEval Entity Extraction Task

PolEval is an example of nested named entity recognition tasks. Participants were asked to train their models on 1M subcorpus of the National Corpus of Polish, consisting of around 87k entities with 14 distinct types in 86k sentences.

20.4	persName
13.2	persName.forename
13.0	persName.surname
11.8	orgName
8.4	placeName.settlement
8.1	placeName.country
4.7	geogName
4.5	date
1.0	persName.addName
0.9	placeName.region
0.6	time
0.4	placeName.region
0.3	placeName.district
0.1	placeName.bloc
87.4	*(in total)*

Table 2: Entity types and their respective frequencies (thousands) in 1M subcorpus of the National Corpus of Polish.

Figure 3 presents overlaps of named entities within 1M subcorpus of the National Corpus of Polish. Values are calculated as frequency of both labels overlaps to the frequency of vertical label, eg. *persName.forename* overlaps with *persName* whenever the first one is present, but *persName* overlaps *forename* in only 64% of cases it appeared in the training set (in this case it reflects the fact that all the *persName.forename* are nested in corresponding *persName* but only some of the *persName*s contain *forename*).

In addition to nested named entities, the mentioned dataset contains a marginal number of non-continuous name entities, such as in *gmina miejska Gdynia* 'Gdynia

Municipality' where the single entity is formed from the first and the last words, with the middle one omitted.

These were intentionally ignored. In general, the tested solutions were selected with the assumption that the final test set will share a similar distribution of entity types, overlapping and related problems.

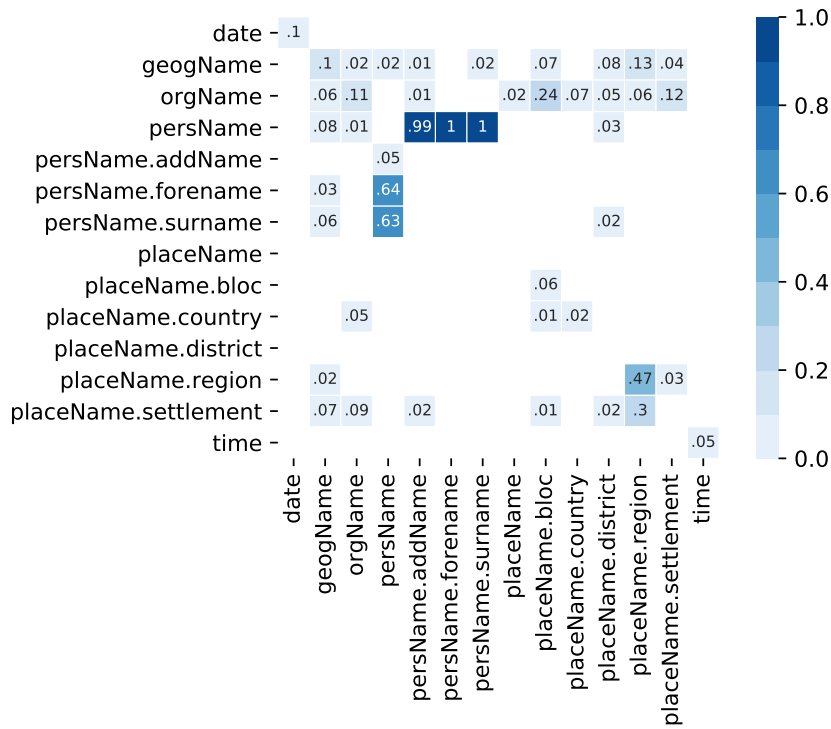

Figure 3: Overlaps of named entities within 1M subcorpus of the National Corpus of Polish. Values calculated as frequency of both labels overlaps to the frequency of vertical label.

2. Towards Choosing an Optimal Solution

Subcorpora described in the previous section was divided into new train (80k sentences), dev and tests sets (both ca. 3k sentences), that were used to start an internal challenge within the local instance of an open source, git-based Gonito.net platform (Graliński et al. 2016). Span F_1 mentioned in the present section is the result of evaluation on so-created, local test set, calculated with the use of *geval* tool. After the official results were published, the submissions described in this paper were uploaded

to an open instance of Gonito.net platform, where all the readers are encouraged to compete.[3]

Most of the solutions rely on training the separate models per (almost) non-overlapping entity groups, that is groups guaranteeing that individual entities within will not collide with each other. Whenever possible, groups consisted of neighboring entities in order to exploit the potential of linear CRF chain. Groups distinguished were (cf. Figure 3 for justification):

— *geogName, placeName*,

— *orgName*,

— *persName.addName, persName.forename, persName.surname*,

— *persName*,

— *placeName.bloc, placeName.region, placeName.country, placeName.district*,

— *time, date, placeName.settlement*.

This approach excludes the possibility of nesting the same type of named entity by design, ignoring that eg. half of *placeName.region* objects have a lower-level *placeName.region* inside. The problem was intentionally left for further exploiting, bearing the expected classes' popularity and limited time in mind.

2.1. Baseline: Search-Based Structured Prediction

As a baseline we decided to rely on the search-based structured prediction, an effective algorithm for reducing structured prediction problems to classification problems (Daumé III et al. 2009), implemented in the *Vowpal Wabbit* machine learning system.[4] Training was performed in 3 passes, with copying features from neighboring lines and search history length set to 6, utilizing the following features:

— token length;

— whether token contains: uppercase letter, lowercase letter, digits, punctuation, dash, colon, only digits, only uppercase letters, only lowercase letters, only punctuation;

— if token was found on the predefined list of first names, surnames, towns, communes, streets, institutions, music bands, geographical names and countries (sourced from Wikipedia, TERYT database and Rymut's dictionary (Rymut 1992));

[3] See: https://gonito.net/challenge/poleval-2018-ner

[4] https://github.com/JohnLangford/vowpal_wabbit

— character n-grams (ranging from 4 to 6) and distinguished affixes,

— rough representation of the token, eg. `Aa+` for *Adam*, `A+` for *NASA* and `9+#9+` for *20:27*;

— effect of analysis with *LanguageTool*, namely: length of lemma, affixes, lemma, morphological tags.

The system described above was able to achieve a span F_1 of 0.82 on test set ({4a1327}[5]).

2.2. LM-LSTM-CRF

The first neural approach tested was based on LM-LSTM-CRF sequence labeling tool[6], implementing the method proposed by Liu et al. (2018), where a character-level language model is trained at the same time, in addition to the sequence labeling model (note that in this method LM is not pre-trained on a large corpus, but trained only on the task data, which is one of the distinguishing features when compared to contextual string embeddings (Akbik et al. 2018)).

For the purposes of using the method, GloVe embeddings (Pennington et al. 2014) were trained on a very large, freely available[7] Common Crawl-based Web corpus of Polish (Buck et al. 2014). After basic filtering, tokenization was performed with *toki* utility (Radziszewski and Śniatowski 2011), because it is distributed along with compatible SRX rules mimicking the standard can be found in the National Corpus of Polish. After postprocessing, the corpus consisted of 27 354 330 800 tokens, 119 330 367 of which were unique. Embeddings were generated for all the tokens present in PolEval task's corpora (symmetric, cased, 300 dimensions, 30 iterations, window size of 15).

The best-performing models of this type were trained for 100 epochs, with the default settings (except higher dimension of word embeddings and disabled word embedding fine tuning), achieving a span F_1 of 0.87 on our test set, outperforming baseline by 5 percentage points ({f2c8fc}).

2.3. Contextual String Embeddings

Contextual String Embeddings were proposed by Akbik et al. (2018), who showed that the internal states of a trained character language model can be used to create

[5] This is the reference code to a repository stored at Gonito.net. The repository may be also accessed by going to `http://gonito.net/q` and entering the code there.

[6] `https://github.com/LiyuanLucasLiu/LM-LSTM-CRF`

[7] `http://data.statmt.org/ngrams/raw/`

word embeddings able to outperform the previous state-of-the-art in sequence labeling tasks. The method was implemented in Flair framework[8] we used for the purposes of training the best-performing models.

Forward and backward character-level language models were trained on 1B words corpus of Polish composed in one third of respectively subsamples from: Polish Wikipedia, PolEval's language modeling task (supposedly the National Corpus of Polish) and Polish Common Crawl. The text was tokenized using the same pipeline as in the preparation of GloVe embeddings described above. Subsamples of Wikipedia and PolEval tasks were selected randomly, whereas those sentences were selected from Common Crawl which were characterized by the highest similarity to PolEval sample, as expressed with cross-entropy (Moore and Lewis 2010).

We used exactly the same parameters, settings and assumptions as Akbik et al. (2018), achieving the final perplexity of 2.44 for forward and 2.47 for backward LM.

The final LSTM-CRF sequence labeling models were trained with one bidirectional LSTM layer and 512 hidden states on 300-dimensional GloVe embeddings (cf. the previous section), as well as embeddings from forward and backward LMs with 2048 hidden states. No progress in terms of span F_1 measured on dev set was observed after 30 epochs which distinguishes the method from LM-LSTM-CRF approach. As expected, the models outperformed previous neural solution achieving F-score of 0.88 on the internal test set ({82e4d1}). The submitted models, trained with our dev set included, performed even better, resulting in F-measure about 0.89.

PolEval nested NER task was evaluated in a different manner, combining weighted measures calculated for overlap and exact matches, giving strong premium for the former. The official, final score turned out to be 0.866, compared to 0.851 for the second best and 0.810 for the third.

Code and models accompanying the paper, which can be used to reproduce the results are publicly available at: https://github.com/applicaai/poleval-2018.

3. Discussion

The described solutions and settings were not the only ones tested, eg. 300-dimensional fastText embeddings provided by Grave et al. (2018) were considered, but we found the GloVe ones better suiting the task. Moreover, the Layered-LSTM-CRF[9] was examined, but the results achieved were disappointing, when following the detection order rule proposed by authors, even when contextual string embeddings

[8] https://github.com/zalandoresearch/flair
[9] https://github.com/meizhiju/layered-bilstm-crf

were used. It may be due to the specific character of the attempted dataset, where given two entity classes it is not known which one will appear in inside and which in outside layers. Since this approach was not sufficiently tested due to the lack of time, we are not reporting it in details.

Furthermore, the layered-LSTM inspired method was tested for second-order LSTM-CRF models whenever it could be beneficial, especially for *persName* tag, that should appear outside every, lower-lever classes group (*persName.forename*, *persName.surname*, *persName.addName*). Including information about those had no impact on the overall performance despite substantially affecting learning speed.

After the predicted answers were sent, LM training continued until no progress was observed, achieving the final perplexity of 2.41 for the forward and 2.46 for the backward model. This encouraged us to test how it could affect the overall results. However, no improvement of sequence labeling model was observed, and the only change was a steeper learning curve (the same accuracy was achieved after fewer epochs).

References

Akbik A., Blythe D. and Vollgraf R. (2018). *Contextual String Embeddings for Sequence Labeling*. [in:] *COLING 2018, 27th International Conference on Computational Linguistics*, pp. 1638–1649.

Bojanowski P., Grave E., Joulin A. and Mikolov T. (2017). *Enriching Word Vectors with Subword Information*. „Transactions of the Association for Computational Linguistics", 5, pp. 135–146.

Buck C., Heafield K. and van Ooyen B. (2014). *N-gram Counts and Language Models from the Common Crawl*. [in:] *Proceedings of the Language Resources and Evaluation Conference*, Reykjavik, Iceland.

Celikyilmaz A., Hakkani-Tür D., Pasupat P. and Sarikaya R. (2015). *Enriching Word Embeddings Using Knowledge Graph for Semantic Tagging in Conversational Dialog Systems*. [in:] *AAAI Spring Symposium Series*. Association for the Advancement of Artificial Intelligence.

Daumé III H., Langford J. and Marcu D. (2009). *Search-based Structured Prediction*. „Machine Learning Journal".

Graliński F., Jaworski R., Borchmann Ł. and Wierzchoń P. (2016). *Gonito.net – Open Platform for Research Competition, Cooperation and Reproducibility*. [in:] Branco A.,

Calzolari N. and Choukri K. (eds.), *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pp. 13–20, Portoroz, Slovenia.

Grave E., Bojanowski P., Gupta P., Joulin A. and Mikolov T. (2018). *Learning Word Vectors for 157 Languages*. [in:] *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Harris Z. S. (1954). *Distributional Structure*. „WORD", 10(2-3), pp. 146–162.

Huang Z., Xu W. and Yu K. (2015). *Bidirectional LSTM-CRF Models for Sequence Tagging*. „CoRR", abs/1508.01991.

Ju M., Miwa M. and Ananiadou S. (2018). *A Neural Layered Model for Nested Named Entity Recognition*. [in:] *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1446–1459. Association for Computational Linguistics.

Katiyar A. and Cardie C. (2018). *Nested Named Entity Recognition Revisited*. [in:] *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 861–871. Association for Computational Linguistics.

Lample G., Ballesteros M., Subramanian S., Kawakami K. and Dyer C. (2016). *Neural Architectures for Named Entity Recognition*. „CoRR", abs/1603.01360.

Liu L., Shang J., Xu F., Ren X., Gui H., Peng J. and Han J. (2018). *Empower Sequence Labeling with Task-aware Neural Language Model*. [in:] *AAAI*.

Moore R. C. and Lewis W. (2010). *Intelligent Selection of Language Model Training Data*. [in:] *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pp. 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nadeau D. and Sekine S. (2007). *A Survey of Named Entity Recognition and Classification*. „Linguisticae Investigationes", 30(1), pp. 3–26. Publisher: John Benjamins Publishing Company.

Pandey P., Pudi V. and Shrivastava M. (2017). *Injecting Word Embeddings with Another Language's Resource: An Application of Bilingual Embeddings*. [in:] *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 116–121. Asian Federation of Natural Language Processing.

Pennington J., Socher R. and Manning C. D. (2014). *Glove: Global Vectors for Word Representation*. [in:] *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Peters M. E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. and Zettlemoyer L. (2018). *Deep Contextualized Word Representations*. „CoRR", abs/1802.05365.

Przepiórkowski A., Bańko M., Górski R. and Lewandowska-Tomaszczyk B. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN.

Radziszewski A. and Śniatowski T. (2011). *Maca — a Configurable Tool to Integrate Polish Morphological Data*. [in:] *Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation*.

Rymut K. (1992). *Słownik nazwisk współcześnie w Polsce używanych*. Instytut Języka Polskiego Polskiej Akademii Nauk.

Young T., Hazarika D., Poria S. and Cambria E. (2017). *Recent Trends in Deep Learning Based Natural Language Processing*. „CoRR", abs/1708.02709.