Liner2 – a Generic Framework for Named Entity Recognition

Micha Marcińczuk and **Jan Kocoń** and **Marcin Oleksy**
Research Group G4.19
Wrocław University of Science and Technology
michal.marcinczuk@pwr.edu.pl
jan.kocon@pwr.edu.pl,marcin.oleksy@pwr.edu.pl

Abstract

In the paper we present an adaptation of Liner2 framework to solve the BSNLP 2017 shared task on multilingual named entity recognition. The tool is tuned to recognize and lemmatize named entities for Polish.

1 Introduction

Liner2 (Marcińczuk et al., 2013) is a generic framework which can be used to solve various tasks based on sequence labeling, i.e. recognition of named entities, temporal expressions, mentions of events. It provides a set of modules (based on statistical models, dictionaries, rules and heuristics) which recognize and annotate certain types of phrases. The framework was already used for recognition of named entities (different levels of granularity, including boundaries, coarse- and fine-grained categories) (Marcińczuk et al., 2012), temporal expressions (Kocoń and Marcińczuk, 2016b) and event mentions (Kocoń and Marcińczuk, 2016a) for Polish.

Task	P [%]	R [%]	F [%]
NER boundaries	86.04	83.02	84.50
NER top9	73.73	69.01	71.30
NER n82	67.65	58.83	62.93
TIMEX boundaries	86.68	81.01	83.75
TIMEX 4class	84.97	76.67	80.61
Event mentions	80.88	77.82	79.32

Figure 1: Precision (P), recall (R) and F-measure (F) for various task obtained with Liner2.

Table 1 contains results for various tasks obtained using Liner2. The results are for strict evaluation. NER refers to recognition of named entity mentions. *NER boundaries* is a model for recognition of named entity boundaries without categorization (Marcinczuk, 2015). The same configuration was used to train a coarse-grained (*NER top9*) and a fine-grained (*NER n82*) model on the KPWr corpus (Broda et al., 2012). The coarse-grained and fine-grained categories are described in Section 2.4.

TIMEX refers to recognition of temporal expression mentions. *TIMEX boundaries* is a model for recognition of temporal expression boundaries without categorization and *TIMEX 4class* is a model for recognition of four classes of temporal expressions: date, time, duration and set (Kocoń and Marcińczuk, 2016b).

The last model named *Event mentions* is for recognition of eight categories of event mentions: action, state, reporting, perception, aspectual, i_action, i_state and light_predicate (Kocoń and Marcińczuk, 2016a). The categorization is done according to the TimeML guideline (Saurí et al., 2006) adopted to Polish language.[1]

2 Solution Description

2.1 Overview

Liner2 processes texts which are tokenized and analyzed with a morphological tagger beforehand. The morphological analysis is optional but it might be useful in some tasks. In case of named entity recognition it has small impact on the results. According to our preliminary experiments on recognition of named entity boundaries the model without base forms and morphological information obtained the value of F-measure lower by only 0.5 percentage point.

After tokenization and morphological analysis the text is passed through a pipeline that consists of the following elements:

[1] https://clarin-pl.eu/dspace/handle/11321/283

Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, pages 86–91,
Valencia, Spain, 4 April 2017. ©2017 Association for Computational Linguistics

1. A statistical model trained on a manually annotated corpus using a Conditional Random Fields modeling (Lafferty et al., 2001). The model uses a rich set of features which are described in Section 2.3.

2. A set of heuristics to merge, group and filter specific categories of named entities according to the BSNLP shared task guidelines.

3. A set of heuristics and dictionaries to lemmatize the named entities.

At this stage, the tool is tuned to recognize named entities for Polish according to the guidelines for the BSNLP 2017 shared task.

2.2 Pre-processing

The input text is tagged using the WCRFT tagger (Radziszewski, 2013) and a morphological dictionary called Morfeusz (Woliński, 2006).

2.3 Features

Liner2 uses the following set of token-level features to represent the input data:

1. **Orthographic features**
    - **orth** – a word itself, in the form in which it is used in the text,
    - n**-prefix** – n first characters of the encountered word form, where $n \in \{1, 2, 3, 4\}$. If the word is shorter than *n*, the missing characters are replaced with '_'.
    - n**-suffix** – n last characters of the encountered word, where $n \in \{1, 2, 3, 4\}$. If the word is shorter than *n*, the missing characters are replaced with '_'. We use prefixes to fill the gap of missing inflected forms of proper names in the gazetteers.
    - **pattern** – encode pattern of characters in the word:
        - ALL_UPPER – all characters are upper case letters, e.g. "NASA",
        - ALL_LOWER – all characters are lower case letters, e.g. "rabbit"
        - DIGITS – all character are digits, e.g. "102",
        - SYMBOLS – all characters are non alphanumeric, e.g. "-_-"',
        - UPPER_INIT – the first character is upper case letter, the other are lower case letters, e.g. "Andrzej",
        - UPPER_CAMEL_CASE – the first character is upper case letter, word contains letters only and has at least one more upper case letter, e.g. "CamelCase",
        - LOWER_CAMEL_CASE – the first character is lower case letter, word contains letters only and has at least one upper case letter, e.g. "pascalCase",
        - MIXED – a sequence of letters, digits and/or symbols, e.g. "H1M1".
    - **binary orthographic features**, the feature is 1 if the condition is met, 0 otherwise. The conditions are:
        - *(word) starts with an upper case letter*,
        - *starts with a lower case letter*,
        - *starts with a symbol*,
        - *starts with a digit*,
        - *contains upper case letter*,
        - *contains a lower case letter*,
        - *contains a symbol*
        - *contains digit.*

      The features are based on filtering rules described in (Marcińczuk and Piasecki, 2011), e.g., first names and surnames start from upper case and do not contain symbols. To some extent these features duplicate the *pattern* feature. However, the *binary features* encode information on the level of single characters (i.e., a presence of a single character with given criteria), while the aim of the *pattern* feature is to encode a repeatable sequence of characters.

2. **Morphological features** – are motivated by the NER grammars which utilise morphological information (Piskorski, 2004). The features are:
    - **base** – a morphological base form of a word,
    - **ctag** – morphological tag generated by tagger,
    - **part of speech**, **case**, **gender**, **number** – enumeration types according to

tagset described in (Przepiórkowski et al., 2009).

3. **Lexicon-based features** – one feature for every lexicon. If a sequence of words is found in a lexicon the first word in the sequence is set as *B* and the other as *I*. If word is not a part of any dictionary entry it is set to *O*.

4. **Wordnet-base features** – are used to generalise the text description and reduce the observation diversity. The are two types of these features:

   - **synonym** – word's synonym, first in the alphabetical order from all word synonyms in Polish Wordnet. The sense of the word is not disambiguated,
   - **hypernym *n*** – a hypernym of the word in the distance of *n*, where $n \in \{1, 2, 3\}$

2.4 Statistical Models

In the pipeline we used two models for named entity recognition: coarse-grained (*NER top9*) and fine-grained (*NER n82*). The coarse-grained model is used to recognize and categorize most of the named entity mentions. The fine-grained model, which has lower recall, is used to change the subcategorization of named entities to conform the BSNLP shared task guideline (see Section 2.5 for more details). Both statistical models were trained on the KPWr corpus (Broda et al., 2012).

The coarse-grained model recognizes the following set of named entity categories:

- event – names of events organized by humans,
- facility – names of buildings and stationary constructions (e.g. monuments) developed by humans,
- living – people names,
- location – names of geographical (e.g, mountains, rivers) and geopolitical entities (e.g., countries, cities),
- organization – names of organizations, institutions, organized groups of people,
- product – names of artifacts created or manufactured by humans (products of mass production, arts, books, newspapers, etc.),
- adjective – adjective forms of proper names,
- numerical – numerical identifiers which indicate entities,
- other – other names which do not fit into previous categories.

The fine-grained model defines more detailed categorization of named entities within the top nine categories. The complete list of named entity categories used in KPWr can be found in *KPWr annotation guidelines – named entities*.[2] The fine-grained model uses a subset of 82 categories and their list can be found in *Liner2.5 model NER*.[3]

2.5 Post-processing

During the post-processing step the following operations are performed:

1. A set of heuristics is used to join successive annotations. According to the guidelines for named entities used in the KPWr corpus nested names are annotated as a sequence of disjoint atomic names. In order to conform the shared task guidelines such names need to be merged into single names.

2. Coarse-grained categories used in the KPWr are mapped onto four categories defined in the shared task. There is a minor discrepancy between KPWr hierarchy of named entity categories and BSNLP categories – names of nations are subtype of organization in KPWr, while in BSNLP shared task they belong to PER category. To overcome this discrepancy we used the fine-grained model to recognize nation names and map them to PER category. Irrelevant for the shared task categories of named entities are discarded, i.e. *adjective*, *numerical* and *other*. The complete set of mapping rules is presented in Table 2.5.

3. Duplicated names, i.e. names with the same form and category, are removed from the set.

The set of heuristics and mapping between categories was defined using the training sets delivered by the organizers of the shared task.

ite

[2] https://clarin-pl.eu/dspace/handle/11321/294

[3] https://clarin-pl.eu/dspace/handle/11321/263

KPWr category	**BSNLP category**
nam_loc	LOC
nam_fac	LOC
nam_liv	PER
nam_org_nation	PER
nam_org	ORG
nam_eve	MISC
nam_pro	MISC
nam_adj	*ignored*
nam_num	*ignored*
nam_oth	*ignored*

Figure 2: Mapping from KPWr categories of named entities to BSNLP categories.

Task	**P**	**R**	**F**
Names matching			
Relaxed partial	66.24	63.27	64.72
Relaxed exact	65.40	62.78	64.07
Strict	71.10	58.81	66.61
Normalization	75.50	44.44	55.95
Coreference			
Document level	7.90	42.71	12.01
Language level	3.70	8.00	5.05
Cross-language level	n/a	n/a	n/a

Figure 3: Results obtained by our system in the Phase I of the BSNLP Challenge for Polish language.

2.6 Lemmatization

To lemmatize named entities we use the following resources:

NELexicon2[4] – a dictionary of more than 2.3 million proper names. Part of the lexicon consists of more than 110k name forms with their lemmas extracted from the Wikipiedia internal links. The links were extracted from a Wikipedia dump using a Pyhon script called *python-g419wikitools*.[5]

Morfeusz SGJP[6] – a morphological dictionary for Polish that contains near 7 millions of word forms. The dictionary was used to retain the plural form of nations names, i.e. „Polacy" (Eng. *Poles*) for „Polaków" (Eng. *Poles* in accusative). After tagging the base form for plural for is a singular form – „Polak" (Eng. *Pole* for „Polacy". According to the BSNLP shared task guidelines the number of the lemmatized form must be the same as in the text. We have extracted all upper case forms with a plural number from the Morfeusz dictionary. The list consists of near 1000 elements.

Algorithm 1 presents the lemmatization algorithm.

3 Evaluation and Summary

Table 3 contains the results obtained by our system in the Phase I of the BSNLP Challenge for Polish language. *Names matching* refers to named entity recognition which was carried out in two ways:[7]

- *Relaxed evaluation: an entity mentioned in a given document is considered to be extracted correctly if the system response includes at least one annotation of a named mention of this entity (regardless whether the extracted mention is base form);*

- *Strict evaluation: the system response should include exactly one annotation for each unique form of a named mention of an entity that is referred to in a given document, i.e., capturing and listing all variants of an entity is required.*

Normalization refers to the named entity lemmatization task. *Coreference* refers to the document-level and cross-language entity matching.

Our system was tuned to recognize and lemmatize named entities only so we did not expect to obtain good results for the coreference resolution tasks. The performance for the strict named entity recognition in terms of precision is similar to our previous results (see *NER top9* in Table 1). However, the recall is significantly lower by more than 10 percentage points. This might indicate that our system does not recognize some of the subcategories of named entities.

At the time of this writing, this system has achieved the top score on the Polish language subtask of the first phase of this Challenge.

[5] `https://clarin-pl.eu/dspace/handle/11321/336`

[7] The description comes from the shared task description: `http://bsnlp-2017.cs.helsinki.fi/shared_task.html`.

Algorithm 1: Lemmatization algorithm.

```
Data: $Name$ – a named entity to lemmatize
$DictMorfPl$ – a dictionary of nominative plural forms with their nominative singular forms from
the Morfeusz SGJP dictionary, e.x.: $Polak \rightarrow Polacy$
$DictPerson$ – a dictionary of people name forms and their nominative forms from NELexicon2.
Parts of the names, i.e. first names and last names, are also included, e.x.:
$JanaNowaka \rightarrow JanNowak$, $Jana \rightarrow Jan$, $Nowaka \rightarrow Nowak$
$DictNelexicon$
Result: $Lemma$ – lemma for the NamedEntity
begin
    $Lemma \longleftarrow NULL$
    /* We use a set of heuristics devoted to PER category.          */
    if Name.type = PER then
        if Name.length = 1 & Name.number = pl & Name.base in DictMorfPl then
            $Lemma \longleftarrow DictMorfPl[Name.base]$
        else if Name.text in DictPerson then
            $Lemma \longleftarrow DictPerson[Name.text]$
        else if Name[0].case = nominative then
            $Lemma \longleftarrow Name.text$
        else
            $Lemma \longleftarrow$ concatenation of bases for each token in Name
    else if Name.base in DictNelexicon then
        $Lemma \longleftarrow DictNelexicon[Name.text]$
    else if Name.length = 1 then
        $Lemma \longleftarrow Name.base$
    else
        $Lemma \longleftarrow Name.text$
```

Acknowledgments

Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

References

Bartosz Broda, Michał Marcinczuk, Marek Maziarz, Adam Radziszewski, and Adam Wardynski. 2012. Kpwr: Towards a free corpus of Polish. In *Proceedings of LREC*, volume 12.

Jan Kocoń and Michał Marcińczuk, 2016a. *Generating of Events Dictionaries from Polish WordNet for the Recognition of Events in Polish Documents*, pages 12–19. Springer International Publishing, Cham.

Jan Kocoń and Michał Marcińczuk. 2016b. Supervised approach to recognise Polish temporal expressions and rule-based interpretation of timexes. *Natural Language Engineering*, pages 1–34.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Michał Marcińczuk and Maciej Piasecki. 2011. Statistical proper name recognition in Polish economic texts. *Control and Cybernetics*, 40:393–418.

Michał Marcińczuk, Michał Stanek, Maciej Piasecki, and Adam Musiał. 2012. Rich set of features for proper name recognition in Polish texts. In *Security and Intelligent Information Systems*, pages 332–344. Springer Berlin Heidelberg.

Michał Marcińczuk, Jan Kocoń, and Maciej Janicki. 2013. Liner2 — A Customizable Framework for Proper Names Recognition for Polish. In Robert Bembenik, Łukasz Skonieczny, Henryk Rybiński, Marzena Kryszkiewicz, and Marek Niezgódka, editors, *Intelligent Tools for Building a Scientific Information Platform*, volume 467 of *Studies in Computational Intelligence*, pages 231–253. Springer.

Michal Marcinczuk. 2015. Automatic construction of complex features in conditional random fields for named entities recognition. In Galia Angelova, Kalina Bontcheva, and Ruslan Mitkov, editors, *Recent Advances in Natural Language Processing,* *RANLP 2015, 7-9 September, 2015, Hissar, Bulgaria*, pages 413–419. RANLP 2015 Organising Committee / ACL.

Jakub Piskorski. 2004. Extraction of Polish named-entities. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*.

Adam Przepiórkowski, Rafał L. Górski, Barbara Lewandowska-Tomaszczyk, and Marek Łaziński. 2009. Narodowy korpus języka polskiego. *Biuletyn Polskiego Towarzystwa Językoznawczego*, 65:47–55.

Adam Radziszewski. 2013. A tiered CRF tagger for Polish. In R. Bembenik, Ł. Skonieczny, H. Rybiński, M. Kryszkiewicz, and M. Niezgódka, editors, *Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions*. Springer Verlag.

Roser Saurí, Jessica Littman, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. TimeML Annotation Guidelines, Version 1.2.1.

Marcin Woliński, 2006. *Morfeusz — a Practical Tool for the Morphological Analysis of Polish*, pages 511–520. Springer Berlin Heidelberg, Berlin, Heidelberg.