Automatic construction of complex features in Conditional Random Fields for Named Entities Recognition

Michał Marcińczuk
Institute of Informatics
Wrocław University of Technology
michal.marcinczuk@pwr.edu.pl

Abstract

Conditional Random Fields (CRFs) have been proven to be very useful in many sequence labelling tasks from the field of natural language processing, including named entity recognition (NER). The advantage of CRFs over other statistical models (like Hidden Markov Models) is that they can utilize a large set of features describing a sequence of observations. On the other hand, CRFs potential function is defined as a linear combination of features, what means, that it cannot model relationships between combinations of input features and output labels. This limitation can be overcome by defining the relationships between atomic features as complex features before training the CRFs. In the paper we present the experimental results of automatic generation of complex features for the named entity recognition task for Polish. A rule-induction algorithm called RIPPER is used to generate a set of rules which are latter transformed into a set of complex features. The extended set of features is used to train a CRFs model.

1 Background

Named entity recognition (NER) is an information extraction task and its goal is to identify and categorize text fragments which refer to some objects. Objects can be referred to by proper names, definite descriptions and noun phrases (LDC, 2008). From the perspective of information extraction tasks proper names are the most valuable as they identify the objects by their unique (to some extends) name. In this paper we will focus on identification of proper names for Polish.

There exist several tools for named entity recognition for Polish, including Liner2[1] (Marcińczuk et al., 2013) and Nerf[2] (Savary and Waszczuk, 2012). So far, the existing tools do not solve the problem of named entity once and for all. For a limited set of named entities (first names, last names, names of countries, cities and roads) the results are 70.53% recall with 91.44% precision (Marcińczuk and Janicki, 2012). Results for a wider range of entities are even lower, i.e. recall of 54% with 93% precision for 56 categories of named entities (Marcińczuk et al., 2013). Savary and Waszczuk (2012) presented a statistical model which obtained 76% recall with 83% precision for names of people, places, organizations, time expressions and name derivations tested on the National Corpus of Polish[3] (Przepiórkowski et al., 2012).

The recent works on named entity recognition focus mainly on improving the machine learning-based approaches. One direction is to decompose the task into two stages: named entity boundary detection and classification (Marcińczuk and Kocoń, 2013). The other is identification of new features which will provide better information to identify the named entities (Marcińczuk and Kocoń, 2013). Another direction is combination of different machine learning methods into a single classifier (Speck and Ngonga Ngomo, 2014). There is also another tendency which is based on increasing the size of training data by their automatic generation from Wikipedia (Al-Rfou et al., 2015). Last but not least direction is improvement of named entity recognition for noisy data, like "tweets" (Piskorski and Ehrmann, 2013; Küçük et al., 2014).

In our study we will follow another route whose goal is to generate a set of complex features based on an existing set of token features. In Section 2

[1] Web page: http://nlp.pwr.wroc.pl/liner2.
[2] Web page: http://zil.ipipan.waw.pl/Nerf.
[3] Home page: http://nkjp.pl

Proceedings of Recent Advances in Natural Language Processing, pages 413–419,
Hissar, Bulgaria, Sep 7–9 2015.

we present the motivation for complex features generation and explain, why the current state-of-the-art approach based on Conditional Random Fields cannot model complex dependences between features and classes on its own. In Section 3 we present a baseline set of features and propose three new auxiliary token features. Section 4 presents a procedure for generation complex features for a predefined set of basic features utilizing an existing algorithm for rule induction called RIPPER. In Section 5 we present the results of empirical evaluation and, finally, in Section 6 we discuss the obtained results.

2 Motivation for complex features

CRFs are type of discriminative models which are trained to maximize the conditional probability of observations (x) and classes (y) sequences $P(y|x)$. The conditional probability distribution is represented as a multiplication of feature functions exponents:

$$P(y|x) = \frac{1}{Z_0} exp\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\lambda_k f_k(y_{i-1}, y_i, x) + \sum_{i=1}^{n}\sum_{k=1}^{m}\mu_k g_k(y_i, x)\right) \quad (1)$$

where Z_0 is a normalization factor, $f_k(y_{i-1}, y_i, x)$ and $g_k(y_i, x)$ are feature functions, and λ_k, μ_k are weights of feature functions which are set during learning process. This probability distribution does not model the relationships between combinations of feature functions and classes. In other words, if a combination of two or more feature functions is a good class indicator, the CRFs will not be able to discover the relationship. However, if the relationship between observation features is known then it can be presented to the CRFs as a set of feature functions. The feature functions which are a combination of two or more observation features will be called complex features. The complex feature functions can be represented as:

$$f'_k(y_{i-1}, y_i, x) = y_{i-1} \circ y_i \circ concat(h_1(x), ..., h_j(x)) \quad (2)$$

$$g'_k(y_i, x) = y_i \circ concat(h_1(x), ..., h_j(x)) \quad (3)$$

where $h_1(x)$, ..., $h_k(x)$ are some observation features. This leads to a conclusion, that the complex dependences between observation features and classes must be predefined in a form of separate feature functions.

To verify the above conclusion we performed the following experiment. Let assume we have a training instance with eight observations ($x_1, ..., x_8$), two observation features h_1 and h_2, and two possible classes A and B. The vectors with observation feature values are presented in Table 1. If we treat every observation as a separate one-element sequence the CRFs model trained with only simple feature functions ($g_k(y_i, x) = y_i \circ h_j(x)$) will not learn to distinguish between classes A and B[4]

x	$h_1(x)$	$h_2(x)$	y
x_1	0	0	A
x_2	0	1	B
x_3	1	0	B
x_4	1	1	A
x_5	0	0	A
x_6	0	1	B
x_7	1	0	B
x_8	1	1	A

Table 1: Feature vectors for observations $x_1, ..., x_8$ and features $h_1(x)$ and $h_2(x)$.

We can observe, that there is a relationship between h_1, h_2 and y, i.e. $y = B$ if $h_1(x) <> h_2(x)$. This relationship can be transformed into a complex function, i.e.:

$$h_3(x) = (h_1 \circ h_2)(x) = concat(h_1(x), h_2(x))$$

If we include the feature $h_3(x)$ (see Table 2) and repeat the training and testing procedure, then the CRFs model will correctly classify the observations. This confirms that the complex dependences between observation features and classes must be beforehand identified and included in the training procedure as a separate set of feature functions.

In the context of named entity recognition tasks the *observation* is a single token. The *class* is a label from a predefined set of labels, i.e. $\{B\text{-}nam, I\text{-}nam, O\text{-}nam\}$, where $B\text{-}nam$ is assigned to tokens starting a named entity, $I\text{-}nam$ is assigned to tokens which are part of a named entity and O is assigned to tokens which are not part

[4] Here we used the CRF++ tool to train and test the model.

x	$h_1(x)$	$h_2(x)$	$h_3(x)$	y
x_1	0	0	00	A
x_2	0	1	01	B
x_3	1	0	10	B
x_4	1	1	00	A
x_5	0	0	00	A
x_6	0	1	01	B
x_7	1	0	10	B
x_8	1	1	00	A

Table 2: Feature vectors for observations $x_1, ..., x_8$ and features $h_1(x)$, $h_2(x)$ and $h_3(x)$.

of any named entity. An *observation feature* is a token attribute, for example an orthographic form, a part of speech or a presence in a gazetteer. A *complex feature* will be a combination of *observation features*, for example *the current token is upper case and the preceding is lower case*.

3 Feature space

3.1 Baseline set of features

The baseline set of features contains features used by Marcińczuk and Kocoń (2013) in recognition of named entities boundaries for Polish. It contains orthographic, morphological, lexicon-based and wordnet-base features. The set contains only one complex feature, i.e. *agreement*. This feature checks the number, case and gender agreement between adjacent tokens.

3.2 New features

Before generating complex features we revised the baseline set of features. After error analysis we have identified three main types of errors which are related to incorrect boundaries detection. The errors are:

- names which are splitted into several tokens which are not separated by white spaces are partially recognized. For example “EX-8.5” (name of an engine model) is splitted into five tokens: *[EX][-][8][.][5]* and only the first token is marked as a named entity, i.e. “*EX*”.
- names which are quoted are partially recognized. For example in “(...) lecture ’New media and social changes’ (...)” only “*New media*” is annotated.
- names in brackets are also partially recognized.

To solve the above problems we introduced three new basic features: *quotation*, *bracket* and *nospace*. The features are described in the following subsections.

3.2.1 The *Nospace* feature

Nospace feature indicates if there is or not a space (or any white space character) between the current and the preceding token.

$$nospace(n) = \begin{cases} 1 & \text{if there is a whitespace character} \\ & \text{between } n-1\text{-th and } n\text{-th tokens} \\ 0 & \text{otherwise} \end{cases}$$

3.2.2 The *Quotation* feature

Quotation feature indicates if the token is between an opening and a closing quotation marks.

$$quotation(n) = \begin{cases} B & \text{if } n\text{-th token is an opening} \\ & \text{quotation mark} \\ I & \text{if } n\text{-th token is between an opening} \\ & \text{and a closing quotation mark} \\ E & \text{if } n\text{-th token is a closing} \\ & \text{quotation mark} \\ O & \text{otherwise} \end{cases}$$

3.2.3 The *Bracket* feature

Bracket feature indicates if the token is between an opening and a closing bracket.

$$bracket(n) = \begin{cases} B & \text{if } n\text{-th token is an opening bracket} \\ I & \text{if } n\text{-th token is between an opening} \\ & \text{and a closing brackets} \\ E & \text{if } n\text{-th token is a closing bracket} \\ O & \text{otherwise} \end{cases}$$

4 Complex feature generation

RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is a rule learning algorithm that can efficiently handle large and noisy datasets. According to Cohen (1995) RIPPER scales nearly linearly with number of examples in a dataset.

We used Java implementation of RIPPER called JRip, which is a part of Weka software (Hall et al., 2009). The set of rules was induced on the tune part of the KPWr corpus (Broda et al., 2012) which contains 62k instances of O class, 3.7k instances of $B - nam$ class and 3k instances of $I - nam$ class. For each token feature we used five features for the adjacent tokens — two preceding tokens, the current token and two following tokens. A sample of token feature vectors for a single feature *orth* is presented in Table 3.

n	orth	orth-2	orth-1	orth-0	orth+1	orth+2
1	Tom	NULL	NULL	Tom	lives	in
2	lives	NULL	Tom	lives	in	Paris
3	in	Tom	lives	in	Paris	NULL
4	Paris	lives	in	Paris	NULL	NULL

Table 3: Token feature vectors for a sample sentence and a single feature *orth* (orthographic form of token)
.

The rule induction process took 2.5 hours on a single 2.4 GHz CPU. The final set of rules consists of 29 rules for *B-nam* class and 24 rules for *I-nam* class. The accuracy of the rules on the tune part was 96.6%. The detailed results are presented in the Table 4.

Class	**P**	**R**	**F**
B-nam	82.5%	79.2%	80.8%
I-nam	86.2%	63.8%	73.3%
O	97.7%	99.1%	98.4%
All	96.6%	96.6%	96.4%

Table 4: Evaluation of the rules on the tune part of the KPWr corpus.

A sample rule generated by JRip is presented on Figure 1. The rule says: *the current token starts a named entity* (B-nam) if *the current token has an upper case letter* (has_upper_case+0 = 1) and *the preceding token does not have only upper case letters* (all_upper-1 = 0) and *the preceding token have only lower case letters* (pattern-1 = ALL_LOWER) and *the following token has an upper case letter* (has_upper_case+1 = 1).

Table 5 contains a list of features which appeared in the rules generated by JRip accompanied with the number of rules containing the feature. The most common features where *has_upper_case* (29 rules), *starts_with_lower_case* (24 rules) and *starts_with_upper_case* (23 rules). These are orthographic features which refer to presence of upper and lower case letters — in Polish upper case letters indicate most of named entity. The new features described in Section 3.2 also appeared in the rules — *parenthesis* and *nospace* appeared in 8 rules and *quotation* in 1 rule. This means that the new features combined with other features are useful in named entity boundary detection.

The set of rules was finally transformed into a set of template features. The transformation consists of removing feature values and keeping only feature names. A feature template for the sample rule from Figure 1 is presented on Figure 2. We use CRF++ [5] implementation of CRFs which generates all possible combinations of feature values for given feature template during the training process. This way CRF++ can explore all combinations of feature values (including the one generated by JRip) and evaluate them in the context of sequence labelling task. The final evaluation of the generated complex features is presented in Section 5.

```
  (has_upper_case+0 = 1)
    and (all_upper-1 = 0)
    and (pattern-1 = ALL_LOWER)
    and (has_upper_case+1 = 1)
  => iobtag=B-nam
```

Figure 1: A sample rule generated by JRip on the tune part of KPWr.

```
has_upper_case:0/all_upper:-1/
  pattern:-1/has_upper_case:1
```

Figure 2: A complex feature converted from the sample rule from Figure 1.

5 Evaluation

We have evaluated three set of features: *baseline* (described in Section 3.1), *baseline with new features* (described in Section 3.2) and *baseline with complex features* (baseline features with new features and automatically generated complex features according to the procedure presented in Section 4).

We decided not to evaluate the set of rules generated by JRip on their own as we did not expect to obtain good results. The performance of the rules on the tune set (set on which the rules were generated) was relatively low and on unseen data it might be even lower.

The evaluation was performed by training CRF-based statistical model using 10-fold cross validation on the train part of the KPWr (see Table 6).

[5] Web page: `http://crfpp.googlecode.com/svn/trunk/doc/index.html`

Feature	Count
has_upper_case	29
starts_with_lower_case	24
starts_with_upper_case	23
ctag	14
pattern	14
agr1	12
class	12
dict_person_first_nam	10
all_upper	9
orth	9
case	8
parenthesis	8
nospace	8
has_lower_case	7
gender	6
length	7
all_alphanumeric	4
all_digits	2
all_letters	3
has_digit	2
number	2
suffix-1	2
struct	2
no_letters	1
prefix-1	1
quotation	1
starts_with_digit	1
suffix-2	1

Table 5: A list of features used to construct the set of rules with a number of rules in which the feature appeared.

We also validate the generality of the feature sets by training the model on the train and tune part of KPWr and testing on the test part of KPWr (see Table 6).

We present results for *strict* and *partial* matching evaluation (Chinchor, 1992). In the *strict* matching the boundaries of recognized annotations must be exactly the same as in the reference corpus. In the *partial* matching the recognition of annotations presence and its boundaries are evaluated separately. This means that annotations which do not exactly match the expected boundaries are treated as partial success.

To check the statistical significance of difference between results we used Student's t-test with a significance level $\alpha = 0.05$ (Dietterich, 1998).

Application of the new three features (*nospace*, *quotation* and *bracket*) improved the F-measure for strict evaluation from 80.30% to 81.13%. The difference is statistically significant for $\alpha = 0.05$ what means that the additional features are useful for the recognition of named entities boundaries. Further improvement was achieved by extending the feature set with the complex features generated with RIPPER algorithm. The F-measure increased to 82.61% and the difference is also statistically significant.

Similar increase of F-measure was observed for the test part of KPWr. The initial value of F-measure increased from 82.40% for baseline set of features to 84.50% for the baseline set of features extended with complex features.

Evaluation	P	R	F
Baseline			
Strict	81.92%	78.74%	80.30%
Partial	88.11%	84.83%	86.44%
Baseline with new features			
Strict	82.79%	79.54%	81.13%
Partial	88.52%	85.22%	86.84%
Baseline with complex features			
Strict	84.10%	81.16%	82.61%
Partial	89.07%	86.25%	87.64%

Table 6: 10-fold cross validation on the train part of KPWr corpus.

Evaluation	P	R	F
Baseline			
Strict	84.25%	80.63%	82.40%
Partial	89.78%	85.80%	87.74%
Baseline with new features			
Strict	84.94%	81.72%	83.29%
Partial	90.22%	86.75%	88.45%
Baseline with complex features			
Strict	86.04%	83.02%	84.50%
Partial	90.73%	87.63%	89.15%

Table 7: Evaluation on the test part of the KPWr corpus.

6 Conclusions

A rule learning algorithms such as RIPPER can be successfully used to improve the performance of a CRF-based statistical model. RIPPER can find a dependences between token features and their classes. The dependences can be expressed as a set of rules which can be latter transformed into a set of feature templates for CRFs.

Despite the improvement we achieved, the final performance of named entity recognition is still far from perfect. There are same possible reasons for that. First of all, the complex features generated by RIPPER have form of conjunction of positive assertions. This means that RIPPER will not produce rules with negation (i.e. if $h_j(x) <>' b'$ then ...). This can be achieved by enumerating all possible values for feature h_j and constructing a set of negated features but this approach might be ineffective due to large space of possible values (especially orthographic and base forms).

The other limitation of this approach is lack of long distance dependences modelling. For example, if a sequence of tokens T in one sentence has labelling L, then there is high probability that the same sequence in an another sentence will have the same labelling. In the current approach there is no linking between the same sequences of tokens.

Also the discrepancy between strict and partial matching evaluation shows, that there is still a problem with proper boundary detection of named entities. This is a problem for long names, like titles which are not quoted. In such cases there is no orthographic indication, where the title ends and its ending is recognized incorrectly.

Acknowledgement Work financed by the Polish Ministry of Science and Higher Education, a program in support of scientific units involved in the development of a European research infrastructure for the humanities and social sciences in the scope of the consortia CLARIN ERIC and ESS-ERIC, 2015-2016

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-NER: Massive multilingual named entity recognition. *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada, April 30 - May 2, 2015*, April.

Bartosz Broda, Michał Marcińczuk, Marek Maziarz, Adam Radziszewski, and Adam Wardyński. 2012. KPWr: Towards a Free Corpus of Polish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of LREC'12*. ELRA.

Nancy Chinchor. 1992. MUC-4 Evaluation Metrics. In *Proceedings of the Fourth Message Understanding Conference*, pages 22–29.

William W. Cohen. 1995. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Dilek Küçük, Guillaume Jacquet, and Ralf Steinberger. 2014. Named entity recognition on turkish tweets. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 450–454. European Language Resources Association (ELRA).

LDC. 2008. ACE (Automatic Content Extraction) English Annotation Guidelines for Relations (Version 6.2).

Michał Marcińczuk and Maciej Janicki. 2012. Optimizing CRF-Based Model for Proper Name Recognition in Polish Texts. In Alexander F. Gelbukh, editor, *CICLing (1)*, volume 7181 of *Lecture Notes in Computer Science*, pages 258–269. Springer.

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.

Michał Marcińczuk and Jan Kocoń. 2013. Recognition of named entities boundaries in polish texts. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 94–99, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jakub Piskorski and Maud Ehrmann. 2013. On named entity recognition in targeted twitter streams in polish. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 84–93, Sofia, Bulgaria, August. Association for Computational Linguistics.

Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. 2012. *Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]*. Wydawnictwo Naukowe PWN, Warsaw.

Agata Savary and Jakub Waszczuk. 2012. Narzędzia do anotacji jednostek nazewniczych. In Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors, *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN. Creative Commons Uznanie Autorstwa 3.0 Polska.

René Speck and Axel-Cyrille Ngonga Ngomo. 2014. Ensemble learning for named entity recognition. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534. Springer International Publishing.