Automatic Construction of a Polish Legal Dictionary with Mappings to Extra-Legal Terms Established via Word Embeddings

Aleksander Smywiński-Pohl
Karol Lasocki
apohllo@agh.edu.pl
karolasocki@gmail.com
AGH University of Science and Technology
Kraków, Poland

Krzysztof Wróbel
Marek Strzała
krzysztof@wrobel.pro
strzalamarek@gmail.com
Jagiellonian University
Kraków, Poland

ABSTRACT

The primary objective of this research is finding correspondence between legal and extra-legal terms in Polish by employing unsupervised methods, such as statistics and word embeddings. We investigate the possibility to construct a legal dictionary automatically by employing statistical methods for identifying the legal terms (including multi-word entities) and then finding correspondence between these terms and extra-legal terminology used by laymen, by employing word embeddings inducing algorithms. We compare two popular libraries word2vec and GloVe in a synthetic experiment showing the superiority of word2vec CBOW negative sampling variant in the described problem.

CCS CONCEPTS

• **Artificial intelligence** → **Natural language processing**; • **Information retrieval** → *Information retrieval query processing*; • **Law, social and behavioral sciences** → *Law*.

KEYWORDS

legal terms, multi-word entities, word embeddings, terminology extraction, conversational systems

ACM Reference Format:
Aleksander Smywiński-Pohl, Karol Lasocki, Krzysztof Wróbel, and Marek Strzała. 2019. Automatic Construction of a Polish Legal Dictionary with Mappings to Extra-Legal Terms Established via Word Embeddings. In *Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19), June 17–21, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3322640.3326727

1 INTRODUCTION

Legal documents, statutory law in particular, are sources of legal language. This language, although shares many features with natural languages, has very specialized vocabulary. The set of terms used in these documents is restricted and natural language phenomena such as polysemy and synonymy are reduced, if not completely eradicated. The downside of this process is the fact that legal vocabulary consists merely of technical terms. As a result, if a person without legal background tries to find a specific regulation or lack thereof, they may fail since they don't know the words that are used by the bill creators.

Traditional legal information systems (LIS) are designed for legal professionals, thus they don't provide any means for simplifying access for laymen. On the other hand, general-purpose search engines that are able to bridge that gap, don't give access to the most accurate and up-to-date results concerning legal regulations. People searching for legal information usually end up reading forum or blog posts, rather than statutory law, even though it is fairly accessible.

A proposed solution for this problem is a LIS equipped with a conversational interface. Such a system would contain the same information as traditional LIS, but will offer help for users that are not acquainted with legal terms. As a requirement, the system has to tell the difference between legal and extra-legal terms. In the first case, the system could offer access to the most important regulations concerning given term (its definition in particular), while in the second it could inform the user that the term doesn't belong to the domain and would offer a list of translations.

The content of the remaining part of the article is following: Section 2 presents the approach towards the construction of the dictionary as well as the determination of the correspondence between the terms, Section 3 describes in detail the steps performed in the process of the dictionary construction, Section 4 describes experiments performed in order to assess the validity of the constructed mapping, Section 5 discusses related work, while Section 6 concludes the article with a summary and prospects for future work.

2 APPROACH

Our goals are the construction of a dictionary of legal terms and a mapping between legal and extra-legal terms. These goals are achieved in the following steps:

(1) At first, a large corpus containing documents from legal and extra-legal domains is compiled.
(2) The corpus is cleaned, tokenized and split into sentences.
(3) The corpus is lemmatized and tagged with parts of speech (POS).

ICAIL '19, June 17–21, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6754-7/19/06…$15.00
https://doi.org/10.1145/3322640.3326727

(4) N-gram counts[1] are computed for the corpus.
(5) Log likelihood ratio (LLR) [3] is computed for bigrams and trigrams in order to identify multi-word entities (MWEs).
(6) A second, small corpus containing the texts of statutory law is compiled. It undergoes the preprocessing described in steps (2) and (3).
(7) LLR is computed for *terms*[2] individually for the bills, using the first corpus as a reference.
(8) A **dictionary of legal terms** is determined for each bill based on the statistical data as well as filtering rules concerning POS tags.
(9) A selection of word embedding inducing algorithms are run on that first corpus. The vectors are computed for words as well for MWEs. Cosine similarity between the term vectors is used to establish **correspondence between legal and extra-legal terms**.

Polish is an inflected language thus lemmatization is a crucial step when counting word occurrences. POS tagging allows for filtering terms that do not form valid legal terms, but are very common in legal documents, e.g. prepositions coupled with nouns, e.g. *w dniu* (*on ... day*). Step 5 is used to identify *multi-word entities* (MWEs). In this case LLR measure is used to compute sequences of words that appear together more frequently than it would stem from their individual probabilities.

Step 7 is used to tell apart *legal* and *extra-legal* terms. Terms that have high correlation with given bill are determined legal. That correlation is once again determined using LLR. The large corpus serves as the source of statistical data concerning term appearances. In step 8 the dictionary of the legal terms is constructed. The terms are filtered based on the POS tags determined in step 6 and the LLR values determined in step 7.

Step 9 is used to find the correspondence between the terms. Dense vector representation of terms allows for finding semantically similar expressions, by computing cosine similarity between their vector representation that correlates with human judgments [13]. Since the terms are split into mutually exclusive sets, it is possible to find the most similar term *in the legal domain* for a term in question (assuming it was identified as extra-legal term). Conversely – it is also possible to identify most similar extra-legal terms for a given legal term.

3 DICTIONARY CONSTRUCTION

3.1 Corpora

As the first part of the large corpus, we use full National Corpus of Polish (NCP) [9, 10]. It includes texts of different genres, such as novels, transcripts of parliamentary speeches and newspaper articles. As the second part of that corpus, we have created a compilation of almost 2 million judgments taken from the following courts: Polish Supreme Court, Polish Constitutional Tribunal, Polish common courts, Polish National Chamber of Appeal, Polish administrative courts. We name it Polish Judgments Corpus (PJC).

The statistics of both parts of the corpus are summarized in Table 1. *Lemma* is understood as a word base form together with the determined POS, thus the number of lemmas is larger than in other sources citing NCP. The legal part of the corpus is almost twice larger than the other in terms of the number of tokens. However, it has 8 times smaller vocabulary and sentences are 3 times longer.

Table 1: The statistics computed for the two parts of the large corpus. NCP = National Corpus of Polish, PJC = Polish Journal Corpus.

Measure	NCP	PJC
Number of tokens	2 591 817 208	4 076 628 858
Number of distinct lemmas	8 423 869	1 051 463
Number of distinct bigrams	119 858 033	27 495 551
Number of distinct trigrams	596 275 165	165 887 557
Size in GBs	8	26
Number of documents	3 911 382	1 944 071
Average sentence length	11.9	36.6

The construction of the second corpus was much more difficult. Polish statutory law is available via ISAP webpage[3]. It contains all bills published in the Journal of Laws of the Republic of Poland (*Pol.* Dziennik Ustaw) since 1918. Yet the only available format of the documents is PDF. Taking into account these obstacles, we have decided to test our approach only on one bill – Polish Penal Code.

3.2 Text Pre-Processing

The pre-processing of the texts was dependent on their source. Texts in NCP need to be cleaned in order to remove (pretty rare) XML markup, such as `<gap>` indicating the fact that a speech transcript was not complete. Some of the texts taken from PJC were already pre-processed, but due to the varied number of sources, some issues, such as broken words or sparse HTML markup were observed. The Polish statutory law is available in PDFs, so the pre-processing started with conversion to plain texts, the broken words at the end of lines were joined in the final step. After initial pre-processing, the texts were broken into tokens and sentences with the help of Maca [11] – a tool dedicated to tokenization and other transformations of Polish texts.

3.3 Lemmatization and Tagging

Pre-processed texts were lemmatized and tagged with POSes with the help of Polish morphological analyzer Morfeusz [16] and the Polish tagger KRNNT [17] – a tagger trained on the NCP corpus (its manually tagged part).

Since Polish is highly inflected language, the tags are compositional in nature – e.g. `subst:nom:sg:m1` means that the words is a noun (`subst`), in nominal (`nom`) and in singular (`sg`) and belongs to first group of masculine nouns (`m1`). The first part of the tag indicates the POS of the token, the rest indicates values of grammatical categories.

In order to identify legal terms, we have decided to keep only the lemma and its POS, stripping away values of the grammatical categories. As a result, inflected forms were substituted with lemmas supplemented by a reduced grammatical information.

[1] A tuple (*lemma*, *POS*) is used as a key and $N \in \{1, 2, 3\}$.
[2] Single words together with MWEs that include at least one noun and fulfill some other restrictions are collectively called *terms*.
[3] http://isap.sejm.gov.pl

3.4 Detection of Multi-Word Expressions

Identification of multi-word expressions was performed with the help of Log-likelihood ratio (LLR) – a measure introduced in [3]. We have counted n-grams up to trigrams using SRILM toolkit [15]. As indicated in Section 3.3, we have substituted the inflected forms of words by their lemma and POS. E.g. a sentence

> *Powołując się na orzecznictwo Sąd Rejonowy stwierdził...*
> (Eng. *Refering to the jurisdiction District Court judged...*)

was substituted with

```
powoływać:pcon się:qub na:prep orzecznictwo:subst
sąd:subst rejonowy:adj stwierdzić:praet
```

and was sent to SRILM to compute the n-gram counts.

In order to identify MWEs consisting of 2 words, for each bigram we have computed the LLR value using the `llr_root` function from the Python implementation of LLR[4] which uses a contingency table to compute the words correlation. This variant of the function tells apart words that are positively and negatively correlated.

Bigrams with high values of LLR are good candidates for multi-word legal terms. An analysis based on the bigrams extracted from the Polish Penal Code showed that most of the expressions have LLR value above 0, thus that value was selected as the threshold. The candidates were then further filtered to only include terms consisting of nouns and adjectives and including at least one noun. The Penal Code included 1532 such terms.

MWEs with 3 tokens were identified in a similar manner, i.e. the same measure was employed. Yet direct application of the procedure is impossible, since a 2x2 contingency table is required. Thus we have treated first two tokens as a single token and restricted the computation only to the trigrams that included at their beginning a bigram identified in the first stage of the procedure as a candidate legal term. Since the constituting bigrams already included at least one noun, no further filtering was necessary. For the Polish Penal Code the procedure gave 131 candidate legal trigrams, with best candidates presented in Table 2. The words underlined in the translation are missing in the source trigram (probably they would be found in higher-order n-grams). As it might be observed four of the terms directly relate to the penalties and the other two are strongly related to the Penal Code (e.g. offences done *in the territory of the Republic of Poland* or *under the influence of a narcotic drug*).

3.5 Identification of Legal Terms

Identification of terms belonging to the legal domain was performed based on the same measure, i.e. LLR. In the contingency table, the rows included counts for the term presence and absence, while the columns included counts for a given statutory law vs. the large corpus. Samples of unigrams of the legal and extra-legal terms identified in the Polish Penal Code are given in Table 3. The terms identified as legal certainly belong to the domain of penal law. On the other hand *Poland* in its common form is judged an extra-legal term, which contrast with *territory of the Republic of Poland* mentioned earlier.

Table 2: Trigrams taken from the Polish Penal Code with the highest values of LLR. The POS tags were removed in order to preserve space. The words that would appear in a tetragram are underlined.

Trigram *English translation*	**LLR**
`wypadek mały waga` *minor offence*	18.4
`granica ustawowy zagrożenie` *limit of the statutory penalty risk*	15.5
`kara pozbawienie wolność` *imprisonment*	15.3
`terytorium rzeczpospolita polski` *territory of the Republic of Poland*	11.1
`górny granica ustawowy` *upper limit of the statutory penalty risk*	11.1
`wpływ środek odurzający` *influence of a narcotic drug*	10.3

Table 3: Legal and extra-legal unigrams identified using LLR measure in the Polish Penal Code.

Unigram	English translation	LLR
	Legal terms	
`kara:subst`	penalty	85.4
`wolność:subst`	liberty	77.4
`przestępstwo:subst`	crime	52.4
`sprawca:subst`	perpetrator	49.8
`czyn:subst`	act	41.2
	Extra-legal terms	
`komisja:subst`	commission	-4.0
`polska:subst`	Poland	-4.1
`mecz:subst`	match	-4.5
`dom:subst`	house	-4.7
`co:subst`	something	-9.8

3.6 Legal and Extra-Legal Terms' Correspondence

In the last stage of the dictionary construction, we have established the correspondence between legal and extra-legal terms by employing word embeddings – vectors of relatively small size (100-600) containing real values. A number of algorithms [6–8] was devised in order to produce such vectors. The resulting vectors are called *word embeddings* and they proved their utility in a number of NLP tasks ([2, 5, 14]).

In our experiments we have used GloVe [8] and word2vec [7] to compute the word embeddings. The large corpus was used to induce the word embeddings. Before the algorithms were run, the spaces dividing the bigrams and trigrams identified as legal were substituted with underscores. Thus these terms were treated as single tokens by the algorithm and the vector representation was computed for the whole expression.

[4] https://github.com/tdunning/python-llr

Table 4: Examples of terms from legal domain and their extra-legal counterparts established using word embeddings computed by word2vec. Sim = cosine similarity.

Legal term	Extra-legal term	Sim.
`zabójstwo` (*homicide*)	`morderstwo` (*murder*)	0.868
`zamach` (*attack*)	`terrorysta` (*terrorist*)	0.712
`kara pozbawienie wolność` (*imprisonment*)	`aresztowanie` (*arrest*)	0.656
`przemoc` (*violence*)	`agresja` (*aggression*)	0.803
`groźba bezprawny` (*illegal threat*)	`szantaż` (*blackmail*)	0.578
`obcy wywiad` (*foreign intelligence*)	`szpieg` (*spy*)	0.615

To find terms related to a given term the cosine similarity between the vectors representing the terms is computed. The set of 100 most similar terms according to this measure are retrieved. Since each term is assigned either to the legal or extra-legal group, finding the corresponding term means finding the most similar term from the opposite group. Table 4 includes some examples with terms from the legal domain and their corresponding terms outside of the domain computed according to the described procedure. The synonymy relation is established if the source legal term appears as the most similar legal term for the corresponding extra-legal term.

4 EXPERIMENTS

In the experiment we wanted to find out which of the most popular word embeddings is best suited for establishing the correspondence between legal and extra-legal term. The dimension of the vectors was set to 100 and the minimal term count was set to 5. We tested the following word embedding models:

- GloVe – symmetric window of size 15, 200 training iterations (GloVe),
- word2vec – symmetric window of size 8, Continuous Bag of Words (CBOW), 50 training iterations, negative sampling (w2v ns),
- word2vec – symmetric window of size 8, CBOW, 50 training iterations, hierarchical softmax (w2v hs).

We have extracted more than 2300 legal terms from the Polish Penal Code by applying the procedure described in Section 3. To automatically assess the validity of the corresponding terms we have employed the following procedure: for each term identified as legal (T_l) we identified the corresponding extra-legal term (T_{el}) using the cosine similarity between word embeddings and then identified the corresponding legal term $\hat{T}_l$ using the same measure. If T_l and $\hat{T}_l$ were the same (i.e. there was a cycle from legal term to extra-legal term and back to the same legal term), we assumed that there is a strong correspondence between the terms. Manual inspection of a sample of such cycles showed that this assumption is valid. To measure the precision of the mappings we employed mean average precision (MAP), measure popular in information retrieval. In general the precision of the mapping was computed according to the position of the T_l among the terms corresponding to T_{el} – we assumed that only T_l corresponds to T_{el}, so $MAP = \frac{100}{position}$, even though there could be other strongly related terms among the $position - 1$ terms with higher cosine similarity to T_{el}.

Experiments have been performed on two sets of legal terms: automatically chosen terms described in Section 3.5 (**LegalA**) and 300 terms manually chosen by a lawyer (**LegalM**). The second set of terms was not comprehensive, but was strongly related to the penal law.

Figure 1 shows results of our experiment. The first row contains the results for the LegalA set, while the second for the LegalM set. The first column reports the MAP for unigrams, bigrams, trigrams and in total for each word embedding model. It is evident that MAP is decreasing with the length of the n-grams and it is probably caused by smaller frequency of higher-order n-grams and less training data thereof.

The second column plots MAP against a minimal *cosine distance* (opposite of cosine similarity, i.e. $1-cos(\theta)$) between the corresponding terms. It is observed that MAP decreases with the increasing cosine distance, but this might be better observed for the much larger LegalA set. What is also evident for the larger set is the superiority of word2vec model, negative sampling in particular, especially when the distance grows beyond 0.3. The function between the distance and MAP is almost linear for the LegalA set in the range 0.2-0.5. This has practical implications, since the distance allows to easily control the system based on such a dictionary, protecting it from proposing misleading suggestions for an uniformed user.

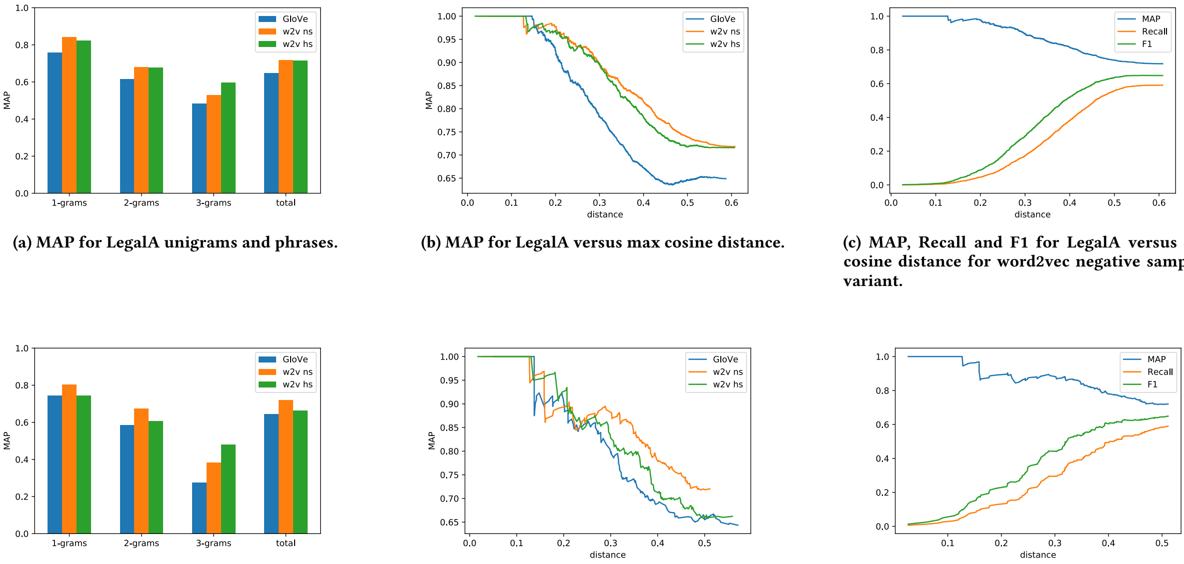
The last column shows MAP together with Recall (measured as the number of valid $\hat{T}_l$ terms that are within the cosine distance) and F1 (harmonic mean of MAP and Recall) for the best performing word2vec negative sampling variant.

5 RELATED WORK

To the best of our knowledge the problem described in our paper (i.e. automatic construction of a legal dictionary including correspondence between legal and extra-legal terms) was not tackled earlier. Regarding the classification of legal terms [12] proposes three groups: pure legal concepts, not used outside of law; legal terminology found in everyday speech and everyday words that have special meaning in law. [1] adopts that view in the problem of terminology translation. Regarding analysis of Polish legal documents [4] describes the application of RAKE keyword extraction algorithm in the legal domain.

The primary difference between our approach is that keyword extraction and dictionary building are related, yet different problems. Keyword extraction is best suited for the long documents, such as judgments which have to be concisely described, while dictionary construction is more concerned with compendia, such as the Penal Code and are less limited in scope. Moreover the authors do not provide any manual or synthetic test for measuring the performance of their method, so it is not possible to compare with their results.

Regarding our algorithm which is based only on distinction between legal and extra-legal terms, we have to stress, that we are primarily concerned with the terminology found in the statutory law, since the proposed solution works in the context of a legal information system. Thus only the terms found in the bills (pure legal concepts) are concerned.



(a) MAP for LegalA unigrams and phrases.

(b) MAP for LegalA versus max cosine distance.

(c) MAP, Recall and F1 for LegalA versus max cosine distance for word2vec negative sampling variant.

(d) MAP for LegalM unigrams and phrases.

(e) MAP for LegalM versus max cosine distance.

(f) MAP, Recall and F1 for LegalM versus max cosine distance for word2vec negative sampling variant.

Figure 1: Results of the experiments.

6 CONCLUSIONS

We have presented a method of building a legal dictionary with a mapping of legal terms to extra-legal terms. Among the tested word embedding models word2vec CBOW with negative sampling seems to be best suited for the construction of such a dictionary. Yet there is much work that has to be done. First of all, we plan to employ more systematic manual validation of the obtained results, especially covering other statutory law, with the help of legal professionals. Then we plan to measure the usefulness of the mappings for people not well acquainted with law, by introducing dictionary driven dialog in the tests. And the last but not the least we want to check if it is possible to automatically classify the semantic relations between the legal terms by employing relation extraction methods.

6.1 Acknowledgments

This work was supported by the Polish National Centre for Research and Development – LIDER Program under Grant LIDER/27/0164/L-8/16/NCBR/2017 titled "Lemkin – intelligent legal information system" and in part by PLGrid Infrastructure.

REFERENCES

[1] Enrique Alcaraz and Brian Hughes. 2014. *Legal translation explained*. Routledge.

[2] Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 740–750.

[3] Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics* 19, 1 (1993), 61–74.

[4] Michał Jungiewicz and Michał Łopuszyński. 2014. Unsupervised keyword extraction from Polish legal texts. In *International Conference on Natural Language Processing*. Springer, 65–70.

[5] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[6] Rémi Lebret and Ronan Collobert. 2013. Word emdeddings through hellinger PCA. *arXiv preprint arXiv:1312.5542* (2013).

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[9] Piotr Pęzik. 2012. Wyszukiwarka PELCRA dla danych NKJP. In *Narodowy Korpus Języka Polskiego*, Adam Przepiórkowski, Mirosław Bańko, Rafał Górski, and Barbara Lewandowska-Tomaszczyk (Eds.). Wydawnictwo Naukowe PWN, 253–279.

[10] Adam Przepiórkowskis, Mirosław Bańko, Rafał Górski, and Barbara Lewandowska-Tomaszczyk. 2012. *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN.

[11] Adam Radziszewski and Tomasz Śniatowski. 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*.

[12] Alison Riley. 1996. The meaning of words in english legal texts: Mastering the vocabulary of the law—a legal task. *The Law Teacher* 30, 1 (1996), 68–83.

[13] Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 1025–1036.

[14] Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076* (2016).

[15] Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

[16] Marcin Woliński. 2006. Morfeusz – a practical tool for the morphological analysis of Polish. In *Intelligent Information Processing and Web Mining*. Springer, 511–520.

[17] Krzysztof Wróbel. 2017. KRNNT: Polish Recurrent Neural Network Tagger. In *Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Zygmunt Vetulani and Patrick Paroubek (Eds.). Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu, 386–391.
]