Morphosyntactic disambiguation and segmentation for historical Polish with graph-based conditional random fields

Jakub Waszczuk, Witold Kieraś, Marcin Woliński

Morphosyntactic disambiguation and segmentation for historical Polish with graph-based conditional random fields

Jakub Waszczuk[1], Witold Kieraś[2], and Marcin Woliński[2]

[1] Heinrich Heine University Düsseldorf, Germany
waszczuk@phil.hhu.de
[2] Institute of Computer Science, Polish Academy of Sciences
{wkieras,wolinski}@ipipan.waw.pl

Abstract. The paper presents a system for joint morphosyntactic disambiguation and segmentation of Polish based on conditional random fields (CRFs). The system is coupled with Morfeusz, a morphosyntactic analyzer for Polish, which represents both morphosyntactic and segmentation ambiguities in the form of a directed acyclic graph (DAG). We rely on constrained linear-chain CRFs generalized to work directly on DAGs, which allows us to perform segmentation as a by-product of morphosyntactic disambiguation. This is in contrast with other existing taggers for Polish, which either neglect the problem of segmentation or rely on heuristics to perform it in a pre-processing stage. We evaluate our system on historical corpora of Polish, where segmentation ambiguities are more prominent than in contemporary Polish, and show that our system significantly outperforms several baseline segmentation methods.

Key words: word segmentation, morphosyntactic tagging, historical Polish, conditional random fields

1 Introduction and related work

Despite the arguments raised in favor of performing end-to-end evaluation of Polish taggers rather than evaluating their disambiguation components only [14], the problem of word-level segmentation in Polish received little attention to this day. This is clearly due to relatively low frequency of segmentation ambiguities in Polish and, consequently, low influence of the phenomenon on tagging accuracy.

Several techniques of morphosyntactic tagging for Polish have been explored over the years, including trigrams [4], transformation-based methods[3] (TaKIPI [12]; Pantera [1]), conditional random fields (WCRFT [13]; Concraft [20]), and neural networks (Toygger [9]; KRNNT [22]; MorphoDiTa-pl [19]). The latter now obtain state-of-the-art results[4] in the task of morphosyntactic tagging for Polish [7]. All these taggers adopt a pipeline architecture, where morphosyntactic

[3] Based on algorithms involving automatic extraction of rules.
[4] See: http://poleval.pl/index.php/results/

disambiguation (including guessing) is preceded by sentence segmentation, word segmentation, and morphosyntactic analysis (not necessarily in this order).[5]

For instance, WMBT, WCRFT, Concraft, and KRNNT all relegate the three "subsidiary" preprocessing tasks to Maca [15]. For word segmentation, Maca relies on ad-hoc conversion rules, which transform and simplify the graph. If segmentation ambiguities persist, simple heuristics – e.g. choosing the shortest path among the remaining segmentation paths – are employed in the end. Another solution is used in MorphoDiTa-pl, which encodes all segmentation ambiguities as morphosyntactic ambiguities. More precisely, it relies on an expanded tagset and conversion routines which allow to encode a given segmentation DAG as a sequence over the expanded tagset. Other Polish taggers seem to neglect the problem of ambiguous segmentation altogether. Toygger, for instance, simply requires that the input text is already segmented and analyzed.

The issue with the existing solutions for Polish is that they assume that word segmentation is performed in preprocessing to morphosyntactic disambiguation. However, neither ad-hoc conversion rules nor simple heuristics are sufficient to deal with segmentation ambiguities, as the latter can require contextual information to be correctly dealt with. The method used in MorphoDiTa-pl actually avoids this pitfall to a certain extent, since it represents segmentation ambiguities in terms of morphosyntactic ambiguities. However, it relies on rather ad-hoc conversion routines which do not seem easily generalizable. One might want to enrich segmentation graphs to account for spelling errors, or to represent several segmentation hypotheses arising in a speech processing system, and it is hard to imagine how conversion routines could account for that.

The problem of word segmentation naturally received more attention for languages where it is more prevalent, such as Chinese or Japanese. Within the context of Chinese, segmentation is often regarded as a labeling task over sequences, where one of two labels – `Start` or `NonStart` – is assigned to each character in the sequence. CRFs, neural networks, and other labeling methods can be then used to discriminate between the possible `Start`/`NonStart` sequences for a given sentence, each sequence uniquely representing the corresponding segmentation [11,3]. The idea of modeling morphological segmentation graphs directly with CRFs was proposed by [10] for Japanese, where a DAG-based CRF assigns a probability to each path in a given segmentation DAG, thus allowing to discriminate between different segmentations and the corresponding morphological descriptions at the same time.

In this work, we use a method similar to [10] and apply it to Polish by extending an existing CRF-based tagger, Concraft, to handle ambiguous segmentation graphs (see Sec. 3). The system is coupled with Morfeusz [21], a morphosyntactic analyzer for Polish, which represents both morphosyntactic and segmentation ambiguities in the form of a DAG (see Sec. 2). Finally, we evaluate our system on historical Polish, where segmentation ambiguities are more prominent than in the contemporary language, and show that our system significantly outperforms several baseline segmentation methods (see Sec. 4).

[5] By extension, this holds true also for ensemble taggers, e.g. PoliTa [8].

2 Morfeusz

Similarly to other systems listed in Section 1, we assume that morphological disambiguation is preceded by dictionary lookup providing all possible interpretations of the input text. This task is performed by the morphological analyser Morfeusz 2 [21], which is well suited to processing historical texts. Namely, Morfeusz allows to customize all linguistically sensitive parts of the analysis: inflectional dictionary, rules of segmentation and the tagset. Appropriate adaptation of Morfeusz to 19$^{\text{th}}$ century and Baroque Polish was done by the authors of the corpora we use, see [5] and [6].

Morfeusz accepts the text as a stream of characters, which it splits into tokens and describes each of them as an inflectional form by assigning a lemma and a morphosyntactic tag containing grammatical features of the form, starting with the part of speech. The tokens generated by Morfeusz are words or parts of words (they do not contain spaces). Segmentation in Morfeusz may be ambiguous. For that reason Morfeusz does not represent its output as a flat list, but as a DAG (directed acyclic graph) of morphological interpretations of tokens.

The past tense of Polish verbs has two variants, e.g. *czytałem* and *(e)m czytał* (1st person singular of 'to read'). The latter variant is interpreted by Morfeusz as consisting of two separate inflectional forms, *(e)m* being an auxiliary form of the verb BYĆ 'to be', which is written together with a preceding token. This variant of past tense was readily used in historical Polish, while in the contemporary language it is present only in specific constructions. The auxiliary form takes part in systematic homonymy with historical forms of numerous adjectives ending in *-em*, e.g. *waszem* ('yours' in instrumental or locative case of masculine or neuter gender). This word may be interpreted in ambiguous ways represented by the graph shown in Figure 1. The first token on each path is a form of the adjective WASZ 'your' in various cases and genders (denoted with simplified Morfeusz tags). The second token is the auxiliary form of the verb BYĆ 'to be' used by the past tense. Depending on the context, each of the three alternative segmentation paths may constitute the correct interpretation.

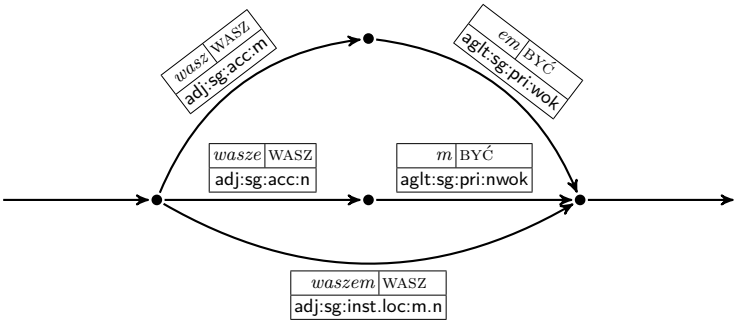

Fig. 1. Ambiguous segmentation of the word *waszem*

Historical Polish provides also examples of accidental ambiguities in segmentation, e.g. the word *potym* can be interpreted as the preposition *po* 'after' written together with the form *tym* of the pronoun TO 'that' or as the form *poty* of the noun POT 'sweat' and an auxiliary *m*.

3 Graph-based CRFs

A sequential CRF [16] defines the conditional probability of a sequence of labels $y \in Y^n$ given a sentence $x \in X^n$ of length n as:

$$p_\theta(y|x) = \frac{\Phi_\theta(y,x)}{Z_\theta(x)} \quad \text{with} \quad Z_\theta(x) = \sum\nolimits_{y' \in Y^n} \Phi_\theta(y',x) \tag{1}$$

Intuitively, the *potential function* $\Phi_\theta(y,x)$ represents the plausibility of sequence of labels y given sentence x – the higher $\Phi_\theta(y,x)$ is, the more probable y w.r.t. x is – while the normalization factor $Z_\theta(x)$ ensures that the probabilities of the individual label sequences sum up to 1. In the particular case of 1-order sequential CRFs, the potential is defined as:

$$\Phi_\theta(y,x) = \exp\Big(\sum\nolimits_{i=1..n} \sum\nolimits_k \theta_k f_k(y_{i-1}, y_i, x)\Big), \tag{2}$$

where θ is a parameter vector and $f_k(y_{i-1}, y_i, x)$ is a binary feature function determining if the k-th feature holds within the context of (y_{i-1}, y_i, x).[6] Defining the exact form of feature functions is a part of the feature engineering process and will depend on the particular application. In our experiments (see Sec. 4), we relied on the Concraft's default feature templates.

Constrained CRFs. Concraft relies on a *constrained* version of sequential CRFs, in which to each position i in the input sequence a set of possible labels $r_i \subseteq Y$ is assigned[7]. When the sets of the potential morphosyntactic interpretations of the individual words in the sentence are available, such position-wise constraints can be successfully applied to both speed up processing and improve the tagging accuracy [20]. Formally, for a given sequence $y \in \prod_i r_i$:

$$p_\theta(y|x,r) = \frac{\Phi_\theta(y,x)}{Z_\theta(x,r)} \quad \text{with} \quad Z_\theta(x,r) = \sum\nolimits_{y' \in \prod_i r_i} \Phi_\theta(y',x). \tag{3}$$

The probability of sequences not respecting the constraints is equal to 0. Note that such sequences are also not accounted for in $Z_\theta(x,r)$.

Constrained DAG-based CRFs. In this work, we rely on a further extension of the constrained model, where the structure of input is a DAG rather than a sequence. Let $D = (N_D, E_D)$ be a segmentation DAG of a given sentence, where N_D and E_D is the set of DAG nodes and edges, respectively. Let also $x_i \in X$ be the word assigned to $i \in E_D$ and $r_i \subseteq Y$ be the set of i's

[6] Intuitively, f_k has a positive influence on the modeled probability if $\theta_k > 0$, negative influence if $\theta_k < 0$, and no influence whatsoever if $\theta_k = 0$.

[7] With $r_i = Y$ for out-of-vocabulary words.

possible labels. We adapt the model to discriminate between the possible paths $y \in P(D, r)$, where $P(D, r)$ denotes the set of labeled paths encoded in D.

$$p_\theta(y|x, r, D) = \frac{\Phi_\theta(y, x, D)}{Z_\theta(x, r, D)} \quad \text{with} \quad Z_\theta(x, r, D) = \sum\nolimits_{y' \in P(D,r)} \Phi_\theta(y', x, D). \quad (4)$$

The potential, in turn, is defined as:

$$\Phi_\theta(y, x, D) = \exp\Big(\sum\nolimits_{i \in Dom(y)} \sum\nolimits_k \theta_k f_k(y_{i-1}, y_i, x, D)\Big), \quad (5)$$

where $Dom(y) \subset E_D$ is the set of edges on the path, y_i denotes the label assigned to edge $i \in E_D$, and y_{i-1} denotes the label assigned to the preceding edge.

Within the context of morphosyntactic tagging, the above model assigns a probability to each DAG-licensed segmentation of the input sentence with a particular morphosyntactic description assigned to each segment on the path. Hence, maximizing $p_\theta(y|x, r, D)$ over all the labeled paths in D jointly performs segmentation and disambiguation, as desired.

Inference. The standard algorithms for sequential CRFs can be straightforwardly adapted to DAG-based CRFs. This includes the *max-product* algorithm used for Viterbi decoding (i.e., finding the most probable labeled path for a given DAG and constraints) and *sum-product* algorithm used for computing the forward and backward sums [18]. These two algorithms, in turn, allow to compute the posterior marginal probabilities of the individual segments and labels in the graph, the expected counts of CRF features per sentence, and to perform the maximum likelihood-based parameter estimation process, neither of which is particularly dependent on the underlying structure (sequence vs. DAG). We refer interested readers to [10] for more information on extending CRFs to DAGs.

Observations. Concraft relies on two types of features: 2-order *transition* features (t_{i-2}, t_{i-1}, t_i), and *observation* features (o_i, t_i), where o_i is an observation (wordform, suffix, prefix, shape, etc.) related to word i. Observations can include information about the preceding and following words – e.g., "the wordform of the segment on position $i-2$" – straightforward to obtain with sequential CRFs. However, in the case of DAGs position $i-2$ may not be uniquely defined.

To overcome this issue, [10] limit the scope of features to two adjacent words, directly accessible in their 1-order model. We adopt a different solution, where the predecessor $i-1$ (successor $i+1$, respectively) of edge $i \in E_D$ is defined as the shortest (in terms of wordform length) edge preceding (following, respectively) i. This allows to define observations in terms of words arbitrarily distant from the current edge, which enables us to use Concraft's feature templates. Note that this does not mean that the model will prefer shorter paths, it simply means that observations are defined at a lower level of granularity. We believe this approach to be reasonable, as long as it is consistently used for both training and tagging.

4 Experimental evaluation

Dataset. Our dataset consists of two separate gold-standard historical corpora of Polish. The first is a manually annotated subcorpus of the Baroque Corpus of

	Baroque	1830-1918
Tagging:		
precision	0.882724	0.903176
recall	0.88303	0.903335
Guessing:		
precision	0.60125	0.610493
recall	0.601214	0.609796
Segmentation:		
precision	0.937455	0.951261
recall	0.948684	0.965946

Segm. baselines	Baroque	1830-1918
shortest path:		
precision	0.712871	0.694111
recall	0.503595	0.517577
longest path:		
precision	0.264848	0.294253
recall	0.41452	0.47628
freq. based:		
precision	0.838571	0.911858
recall	0.724294	0.823025

Table 1. Evaluation (our system on the left, segmentation baselines on the right)

Polish [5] which is still under development at the time of writing. It is currently ca. 430,000 tokens large and consists of samples (ca. 200 words each) excerpted from over 700 documents of various genres published between 1601 and 1772. The other dataset is a 625,000 tokens large manually annotated corpus of Polish texts published between 1830 and 1918 [6]. The corpus consists of samples (ca. 160 words each) excerpted from 1000 documents divided between five genres: fiction, drama, popular science, essays and short newspaper texts. The corpus is balanced according to genre and publication date.

The tagset of the 1830–1918 corpus consists of 1449 possible tags, from which 1292 were chosen at least once by human annotators. The Baroque tagset is much larger: it consists of 2212 possible tags and 1940 of them were used by annotators. The size of the Baroque tagset reflects the extensive time span covered by the corpus as well as significant grammatical changes which took place in that period, such as the grammaticalisation of masculine personal gender. It is assumed that since the turn of the 18^{th} and 19^{th} centuries Polish morphosyntactic system was not subject to major changes.

Evaluation. The results of 10-fold cross-validation of our system on both historical corpora are presented in Tab. 1. We measured the quality of morphosyntactic tagging[8] and segmentation in terms of *precision* and *recall*. If several tags were assigned to a segment in gold data, we considered the choice of our system as correct if it belonged to this set. In case of segmentation, the choices of morphosyntactic tags were not accounted for.

We compared our system with three baseline segmentation methods. The first and the second one systematically chooses the shortest and the longest possible segmentation path, respectively. The third system is based on frequencies with which ambiguous segments are marked as chosen in gold data. Namely, we define the probability $p(x)$ of a segment x as $\#(x \text{ chosen in gold} + 1)/\#(x \text{ present in gold} + 2)$,[9] and the probability of a given segmentation path as a product of the probabilities of its component segments. Our system outperforms all three baseline methods significantly. Best among the baselines, the

[8] Note that these results abstract from the potential morphosyntactic analysis errors.
[9] Increasing all counts by 1 makes the probability of unseed segments equal to 1/2.

frequency-based method suffers from the *length bias* problem, as revealed by the differences between its precision and recall.

5 Conclusions and future work

The existing taggers for Polish either neglect the problem of ambiguous segmentation, or adopt ad-hoc approaches to solve it. By extending an existing CRF-based tagger for Polish, Concraft, to work directly on segmentation graphs provided by Morfeusz, we designed a system which addresses this deficiency by performing disambiguation and word-level segmentation jointly. Evaluation of our system on two historical datasets, both containing a non-negligible amount of segmentation ambiguities, showed that it significantly outperforms several baseline segmentation methods, including a frequency-based method.

The advantages of neural methods, now state-of-the-art in the domain, over CRFs include their ability to capture long-distance dependencies and to incorporate dense vector representations of words. For future work, we would like to explore the possibility of alleviating these weaknesses of CRFs, and the possibility of adapting neural methods to DAG-based ambiguity graphs. Following our claim that contextual information is required to properly deal with segmentation ambiguities, it seems clear that the principal way of improving segmentation accuracy is to focus on the quality of the subsequent NLP modules – disambiguation, parsing – as long as they are able to handle ambiguous segmentations.

6 Acknowledgements

The work being reported was partially supported by a National Science Centre, Poland grant DEC-2014/15/B/HS2/03119.

References

1. Acedański, S.: A morphosyntactic Brill tagger for inflectional languages. In: International Conference on Natural Language Processing. pp. 3–14. Springer (2010)
2. Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.): Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014. ELRA, Reykjavík, Iceland (2014), http://www.lrec-conf.org/proceedings/lrec2014/index.html
3. Chen, X., Qiu, X., Zhu, C., Liu, P., Huang, X.: Long short-term memory neural networks for Chinese word segmentation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1197–1206. ACL (2015), http://www.aclweb.org/anthology/D15-1141
4. Dębowski, L.: Trigram morphosyntactic tagger for Polish. In: Intelligent Information Processing and Web Mining, pp. 409–413. Springer (2004)
5. Kieraś, W., Komosińska, D., Modrzejewski, E., Woliński, M.: Morphosyntactic Annotation of Historical Texts. The Making of the Baroque Corpus of Polish. In: Ekštein, K., Matoušek, V. (eds.) Text, Speech, and Dialogue 20th International

Conference, TSD 2017, Prague, Czech Republic, August 27-31, 2017, Proceedings. LNCS, vol. 10415, pp. 308–316. Springer International Publishing (2017), https://doi.org/10.1007/978-3-319-64206-2_35
6. Kieraś, W., Woliński, M.: Manually Annotated Corpus of Polish Texts Published between 1830 and 1918. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2018. ELRA, Miyazaki, Japan (2018)
7. Kobyliński, Ł., Ogrodniczuk, M.: Results of the PolEval 2017 competition: Part-of-speech tagging shared task. In: Vetulani and Paroubek [17], pp. 362–366
8. Kobyliński, Ł.: PoliTa: A multitagger for Polish. In: Calzolari et al. [2], pp. 2949–2954, http://www.lrec-conf.org/proceedings/lrec2014/index.html
9. Krasnowska-Kieraś, K.: Morphosyntactic disambiguation for Polish with bi-LSTM neural networks. In: Vetulani and Paroubek [17], pp. 367–371
10. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004), http://www.aclweb.org/anthology/W04-3230
11. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics (2004), http://www.aclweb.org/anthology/C04-1081
12. Piasecki, M., Wardyński, A.: Multiclassifier approach to tagging of Polish. In: Proceedings of the International Multiconference on ISSN. vol. 1896, p. 7094
13. Radziszewski, A.: A tiered CRF tagger for Polish. In: Intelligent tools for building a scientific information platform, pp. 215–230. Springer (2013)
14. Radziszewski, A., Acedański, S.: Taggers gonna tag: An argument against evaluating disambiguation capacities of morphosyntactic taggers. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue. pp. 81–87. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
15. Radziszewski, A., Śniatowski, T.: Maca — a configurable tool to integrate Polish morphological data. In: Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation (2011)
16. Sutton, C., McCallum, A., et al.: An introduction to conditional random fields. Foundations and Trends® in Machine Learning 4(4), 267–373 (2012)
17. Vetulani, Z., Paroubek, P. (eds.): Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu, Poznań, Poland (2017)
18. Wainwright, M.J., Jordan, M.I., et al.: Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning 1(1–2), 1–305 (2008)
19. Walentynowicz, W.: MorphoDiTa-based tagger for Polish language (2017), http://hdl.handle.net/11321/425, CLARIN-PL digital repository
20. Waszczuk, J.: Harnessing the CRF Complexity with Domain-Specific Constraints. The Case of Morphosyntactic Tagging of a Highly Inflected Language. In: Proceedings of COLING 2012. pp. 2789–2804 (2012), http://www.aclweb.org/anthology/C12-1170
21. Woliński, M.: Morfeusz reloaded. In: Calzolari et al. [2], pp. 1106–1111, http://www.lrec-conf.org/proceedings/lrec2014/index.html
22. Wróbel, K.: KRNNT : Polish recurrent neural network tagger. In: Vetulani and Paroubek [17], pp. 386–391