Applying Support Vector Machines to POS tagging of the Ainu Language

Karol Nowakowski[*], Michal Ptaszynski[*], Fumito Masui[*], Yoshio Momouchi[**]

[*] Kitami Institute of Technology, 165 Koen-cho, Kitami, Hokkaido 090-8507, Japan
karol_nowakowski@ialab.cs.kitami-it.ac.jp, ptaszynski@cs.kitami-it.ac.jp, f-masui@mail.kitami-it.ac.jp
[**] Professor Emeritus of Hokkai-Gakuen University, Minami 26, Nishi 11, Sapporo 064-0926, Japan

Abstract

We describe our attempt to apply a state-of-the-art sequential tagger – SVMTool – in the task of automatic part-of-speech annotation of the Ainu language, a critically endangered language isolate spoken by the native inhabitants of northern Japan. Our experiments indicated that it performs better than the custom system proposed in previous research (POST-AL), especially when applied to out-of-domain data. The biggest advantage of the model trained using SVMTool over the POST-AL tagger is its ability to guess part-of-speech tags for OoV words, with the accuracy of up to 63%.

1 Introduction

Ainu[1] is a critically endangered language isolate spoken by the native inhabitants of northern parts of Japan. Due to its unique characteristics (such as noun incorporation or the usage of affixes – rather than pronouns – to express grammatical person), it has been the subject of a number of linguistic studies. Nevertheless, it receives little attention in the fields of NLP and Computational Linguistics. There is an ongoing project, started by Nowakowski et al. (2018), to create a large-scale annotated corpus of Ainu, which is expected to trigger further development of language technologies related to Ainu. However, there are few Ainu language experts, which renders the task of manual annotation very time-consuming if not infeasible. A possible solution to the problem is to apply bootstrapping techniques (as described e.g. by Clark et al. (2003)) in order to generate the annotations automatically or semi-automatically. As a starting point for such endeavor, in this paper we describe an experiment comparing the performance of two different automatic POS taggers on Ainu language data.

The remainder of this paper is organized as follows. In Section 2 we shortly describe the characteristics of the Ainu language. In Section 3 we review the related work. In Section 4 we introduce the data used to train the part-of-speech taggers applied in this research. In Section 5, the test data used in evaluation experiments is presented. In Section 6 we explain the modifications to part-of-speech annotations present in the data applied in our experiments and introduce the full POS tagset with statistics. In Section 7 we describe the SVMTool settings used for model generation and tagging process. Section 8 is dedicated to the evaluation experiments and discussion about their results. Finally, Section 9 contains conclusions and ideas for future improvements.

2 Characteristics of the Ainu language

In terms of typology, Ainu is an agglutinative language, with a tendency towards polysynthesis manifested by the presence of such traits as pronominal marking and noun incorporation (especially in the language of classical Ainu literature (Shibatani 1990)). The basic word order is SOV. Ainu verbs – and to lesser extent nouns –

kotan	apapa	ta	a=eponciseanu
kotan	apa-pa	ta	a-e-pon-cise-anu
village	entrance-mouth	at	we/people-for[someone]-small-house-lay

We built a small hut for [her] at the entrance to the village.

Figure 1: Example of polysynthesis in the Ainu language (Tamura 1996).

[1] The word *ainu* (written as *aynu* in modern standard transcription) means "human" and it is also used to refer to the ethnic group in question.

Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages: Vol. 2 Extended Abstracts, pages 17–23,
Honolulu, Hawai'i, USA, February 26–27, 2019.

take a variety of affixes, expressing reciprocity, causativity, plurality and other categories.

History of Ainu as a written language is relatively short. Most documents are transcribed using Latin alphabet and/or Japanese *katakana* script (all textual data used in this research is written in Latin script). Until the last decade of the 20th century there existed no widely accepted standard orthographic rules for the Ainu language[2].

3 Related work

The first and hitherto the only existing part-of-speech tagging tool for the Ainu language was developed by Ptaszynski and Momouchi (2012), under the name POST-AL. It was trained using a dictionary of Ainu compiled by Kirikae (2003) and performed POS disambiguation based on word n-grams obtained from sample sentences included in the dictionary. In 2017, Nowakowski, Ptaszynski and Masui investigated the possibility of improving the system's performance by using two dictionaries instead of one and applying a hybrid method of part-of-speech disambiguation, based on word n-grams and Term Frequency.

Unlike POST-AL, state-of-the-art POS taggers developed for other languages typically utilize part-of-speech annotated language corpora as their training data. One of such tools is the SVMTool by Giménez and Márquez (2004), which is an open source generator of sequential taggers based on Support Vector Machines. It achieves an accuracy of 97.2% in POS tagging of English, but has also been applied in studies dedicated to low-resource languages, such as the ones by Hagemeijer et al. (2014) and Behera et al. (2015).

In this research we carried out an experiment to compare POST-AL and SVMTool. Specifically, we used SVMTool v. 1.3.2 (Perl version)[3] and POST-AL tagger in the variant with hybrid approach to POS disambiguation, which yielded the best results in experiments carried out by Nowakowski, Ptaszynski and Masui (2017).

There are several lexicons of the Ainu language containing information about parts of speech, such as those by Nakagawa (1995), Tamura (1996) and Kirikae (2003). However, the amount of existing POS annotated texts which could be readily applied as a training corpus for a tagging system is negligible. Nowakowski et al. (2018) have included three POS tagged datasets (less than 30 thousand tokens in total) in their corpus. In this research we use one of them – an online dictionary by Bugaeva and Endō (2010) – to produce training data for SVMTool (for details, see the next section).

4 Training data

To train both taggers used in this research, we used the data extracted from A Talking Dictionary of Ainu: A New Version of Kanazawa's Ainu Conversational Dictionary by Bugaeva and Endō (2010), which is an online dictionary based on the *Ainugo kaiwa jiten*, a dictionary compiled by Shōzaburō Kanazawa and Kotora Jinbō, and published in 1898. It contains 3,847 entries.

Apart from isolated headwords, the resource includes 2,459 multi-word items (phrases and sentences) and each of them is annotated with a sequence of POS tags. Using that information, we were able to build a small (12,952 token-tag pairs, excluding punctuation) part-of-speech annotated corpus. A subset of it was excluded from the training data, in order to be used as test data in evaluation experiments (for details, see the next section), which left us with a training corpus of 11,249 token-tag pairs (excluding punctuation).

In order to avoid an increase of Out of Vocabulary words, we decided to retain single-word entries in the training corpus and treated them as separate sentences (by inserting a sentence delimiter after each of them).

The corpus was prepared in column format (one token per line), which is the format accepted by SVMTool. Additionally, for the purpose of applying it with POST-AL, it was converted into a dictionary format, where each entry consists of a token (word or punctuation mark), part-of-speech and a list of sentences the given word appears in (if available). The resulting dictionary contains a total of 2392 entries.

5 Test data

To evaluate the performance of both taggers, we used two sets of held-out data:

TDOA: This dataset consists of 1701 tokens (excluding punctuation) from the A Talking

[2] Standard orthography has been proposed by the Hokkaidō Utari Kyōkai (1994) and is widely used to this day.

[3] The software and its documentation can be downloaded from http://www.lsi.upc.es/~nlp/SVMTool/

Dictionary of Ainu… (Bugaeva and Endō 2010). Samples for the test data were selected in the following way: firstly, all sentences with the token count (excluding punctuation) of 3 and higher were extracted from the training corpus and grouped according to their token count. Secondly, duplicate sentences were eliminated. In the next step, a random sample of 20% was selected from each group. Lastly, the sentences selected for the test data were excluded from the training corpus.

SYOS: Five out of thirteen *yukar* epics included in the *Ainu Shin'yōshū* ("Collection of Ainu songs of gods") by Yukie Chiri (1923). Unlike the A Talking Dictionary of Ainu…, it represents the literary style of Ainu. The text was revised in terms of transcription by an Ainu language expert. It comprises a total of 1606 tokens (excluding punctuation) in 88 sentences.

6 POS annotations and tagset

Before applying the annotations produced by Bugaeva and Endō in our research, we decided to introduce several modifications. All such decisions were consulted with three comprehensive dictionaries including the information about parts of speech, by Nakagawa (1995), Tamura (1996) and Kirikae (2003). We also referred to the classification of word classes proposed by Refsing (1986).

The most notable change is the elimination of two word classes: Numeral (135 occurrences in the original data) and Interrogative (347 occurrences). All tags belonging to these two classes were converted to one of the following tags, depending on morphosyntactic characteristics of words they denote: "Adnoun" (e.g. *sine* – "one [day]") or "Noun" (e.g. *sinep* – "one thing") for Numerals, and "Pronoun" (e.g. *hemanta* – "what"), "Adnoun" (e.g. *inan* – "which"), "Noun" (*hempakniw* – "how many people"), "Adverb" (e.g. *hempara* – "when") or "Locative noun" (e.g. *hunak* – "where") for Interrogatives. The reason for that modification is that, apart from Bugaeva and Endō only Nakagawa classifies such words simply as Numerals and Interrogatives, whereas both Tamura and Kirikae rely on functional criteria in deciding their primary word class. Apart from that, we corrected a number inconsistent annotations and typos, and annotated words for which POS tags were missing in the original data. Moreover, we added three punctuation marks that were absent from the A Talking Dictionary of Ainu…, but often appear in other texts: quotation mark (""), colon (":") and ellipsis ("…").

Tag	Number of occurrences	
	A Talking Dictionary of Ainu… (with modifications)	SYOS
Noun	2799	355
Intransitive verb	2504	297
Transitive verb	1503	174
Personal affix	1114	178
Adverb	1041	65
Conjunctional particle	626	146
Nominalizer	594	36
Locative noun	480	64
Final particle	430	22
Case particle	415	55
Adnoun	343	38
Postpositive adverb	246	8
Verb auxiliary	229	50
Supplementary particle	182	28
Pronoun	166	8
Ditransitive verb	130	18
Complete verb	56	2
Interjection	47	11
Proper noun	47	17
Prefix	0	3
.	3396	55
;	508	0
?	470	12
,	106	102
!	28	14
"	1	50
:	1	1
...	1	2
:--	0	5
Unknown	0	31

Table 1: Complete tagset and statistics.

Gold standard part-of-speech annotation for the SYOS dataset was performed by an Ainu language expert, in accordance with the methodology described by Momouchi et al. (2008).

The complete part-of-speech tagset along with statistics of occurrences in both datasets is presented in Table 1.

7 SVMTool settings

7.1 Model settings

Our model was trained on the column-formatted corpus described in Section 4, with training

parameters set to default values. Appendix A explains the feature set used in each variant of the model applied in this research.

Preliminary experiments revealed that the model assigns tags corresponding to punctuation marks (e.g. ":") to many lexical OoV words. To avoid such behavior, we modified one of the model files containing the list of tags to be considered for OoV tokens, removing such tags from the list.

7.2 Tagging parameters

In the experiments with SVMTool tagger, we investigated the performance with different values of the following parameters[4]:

- Tagging strategy (`-T`) – different strategies apply different tagging schemes (greedy or sentence-level) and different variants of the tagging model are used;
- Tagging direction (`-S`) – LR (left-to-right), RL (right-to-left) or LRL (both directions combined). According to Giménez and Márquez (2012), tagging direction "varies results yielding a significant improvement when both are combined".

8 Results and discussion

Results of POS tagging experiments using SVMTool for each combination of tagging parameters are shown in Tables 3 and 4, while Table 5 presents the results of experiments with POST-AL. Table 2 shows the MFT baselines calculated by SVMTool.

The results indicate that both taggers are better than the baseline and a tagger generated using SVMTool performs better than POST-AL, especially when applied to out-of-domain data (SYOS). The biggest advantage of the model trained using SVMTool is its ability to predict part-of-speech tags for Out of Vocabulary words, which it performs with the accuracy of up to 63% (see Tables 6 and 7), while POST-AL does not have such a mechanism. In fact, if we excluded OoV words from the calculation, in the experiment on SYOS dataset POST-AL would yield slightly higher accuracy than our SVMTool model (1238 versus 1234 correct predictions).

Test data	Accuracy
TDOA	1910 / 2023 (94.41%)
SYOS	1225 / 1847 (66.32%)

Table 2: Most Frequent Tag baseline.

		Direction (`-S`)		
		LR	**LRL**	**RL**
Tagging strategy (`-T`)	0	97.33%	97.08%	89.77%
	1	-	-	90.46%
	2	97.62%	97.23%	90.21%
	4	97.78%	-	-
	5	97.33%	96.89%	90.11%
	6	**97.83%**	-	-

Table 3: Results (Accuracy) of the experiments with SVMTool on TDOA dataset (best result in bold).

		Direction (`-S`)		
		LR	**LRL**	**RL**
Tagging strategy (`-T`)	0	74.93%	74.07%	69.95%
	1	-	-	69.46%
	2	76.99%	76.77%	72.93%
	4	**78.34%**	-	-
	5	75.09%	75.26%	70.06%
	6	78.07%	-	-

Table 4: Results (Accuracy) of the experiments with SVMTool on SYOS dataset (best result in bold).

Test data	Accuracy
TDOA	1939 / 2023 (95.85%)
SYOS	1238 / 1847 (67.03%)

Table 5: Results of the experiments with POST-AL.

Differences in accuracy observed between various tagging strategies offered by SVMTool were also mainly caused by different scores for unknown words, while the results for known words

[4] For details please refer to SVMTool's documentation (Giménez and Márquez 2012).

Category	Accuracy
Known	1950 / 1977 (98.63%)
Unknown	29 / 46 (63.04%)

Table 6: Results of the experiments with SVMTool (`-T 6 -S LR`) on TDOA dataset – Accuracy per category of words.

Category	Accuracy
Known	1234 / 1410 (87.52%)
Unknown	213 / 437 (48.74%)

Table 7: Results of the experiments with SVMTool (`-T 4 -S LR`) on SYOS dataset – Accuracy per category of words.

exhibited much less variance. For instance, in the experiment on SYOS data, the performance for in-vocabulary words was less than 1% higher with the tagging strategy set to `-T 4` as compared to `-T 0` (1234 versus 1224 correct predictions), but at the same time the performance for OoV words improved by over 12% (213 versus 160 correct predictions).

The best performance with both sets of test data was achieved by tagging strategies 4 and 6. According to SVMTool's technical manual (Giménez and Márquez 2012), both of them utilize Model 4 – the variant which addresses the problem of OoV words by artificially marking a portion of the training data as unknown during the learning process. Additionally, tagging strategy 6 maximizes the global (sentence-level) sum of SVM scores, rather than making decisions based on a reduced context.

Contrary to the results reported by Giménez and Márquez (2012), using the combination of both tagging directions (`-S LRL`) did not improve the performance in our experiments – the only case where it yielded slightly higher accuracy than tagging from left to right (`-S LR`) was the experiment on SYOS with tagging strategy set to 5. The reasons behind this behavior shall be investigated in future research.

8.1 Combined accuracy

Apart from using each of the two taggers in isolation, we are also interested in the possibilities of combining them to maximize accuracy. In order to estimate the potential performance of such combination, we calculated to what extent both taggers agree on their output and how accurate those shared predictions are. In the case of SVMTool, we used the predictions with the highest accuracy for each of the two test datasets (i.e. the ones generated with tagging parameters set to `-T 6 -S LR` for TDOA and `-T 4 -S LR`, for SYOS). Results are shown in Table 8.

The accuracy of shared predictions is higher than the total accuracy of either of the two taggers used in isolation. In the future we might leverage this fact to reduce the amount of incorrect annotations when applying both taggers in a cross-training scenario to bootstrap POS annotations for a larger corpus of Ainu texts.

Test data	Common predictions	Shared accuracy
TDOA	1953 / 2023 (96.54%)	1932 / 1953 (98.92%)
SYOS	1317 / 1847 (71.30%)	1196 / 1317 (90.81%)

Table 8: Proportion of common predictions and their accuracy.

9 Conclusions and future work

In this research we used a small amount of part-of-speech annotated Ainu language textual data to train and compare two POS taggers: POST-AL – a system developed specifically for Ainu, based on contextual (n-gram) and statistical (TF) information derived from a lexicon, and a tagger generated using SVMTool – an off-the-shelf generator of sequential taggers based on Support Vector Machines.

Experiments conducted on two different sets of objective data revealed that the SVM based approach is more effective, especially when applied to out-of-domain data, the main reason for higher accuracy being its ability to predict part-of-speech tags for Out of Vocabulary words.

One of the main tasks for the future is to convert other existing Ainu language resources including the information about parts of speech (such as the dictionary by Kirikae (2003)) to a corpus format which could be used with SVMTool or other POS taggers. We also plan to apply POST-AL and SVMTool in a cross-training experiment to bootstrap part-of-speech annotations for a bigger corpus of texts in the Ainu language.

References

Pitambar Behera, Atul Kr. Ojha, and Girish Nath Jha. 2015. *7th Language and Technology Conference, LTC 2015, Poznań, Poland, November 27-29, 2015, Revised Selected Papers*, pp. 393-406.

Anna Bugaeva and Shiho Endō (eds.). 2010. A Talking Dictionary of Ainu: A New Version of Kanazawa's Ainu Conversational dictionary. Retrieved November 25, 2015 from http://lah.soas.ac.uk/projects/ainu/

Yukie Chiri. 1923. *Ainu shin-yōshū* [Ainu songs of gods]. Kyōdo Kenkyūsha, Tokyo

Stephen Clark, James R. Curran and Miles Osborne. 2003. Bootstrapping POS taggers using Unlabelled Data. School of Informatics. University of Edinburgh.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Jesús Giménez and Lluís Màrquez. 2012. *SVMTool: A general POS tagger generator based on Support Vector Machines. Technical Manual v1.4*. TALP Research Center, LSI Department, Universitat Politécnica de Catalunya.

Tjerk Hagemeijer, Michel Généreux, Iris Hendrickx, Amália Mendes, Abigail Tiny, and Armando Zamora. 2014. The Gulf of Guinea Creole Corpora. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pp. 523-529.

Hokkaidō Utari Kyōkai [Hokkaido Ainu Association]. 1994. Akor Itak [Our Language]. Sapporo.

Kotora Jinbō and Shōzaburō Kanazawa. 1898. *Ainugo kaiwa jiten* [Ainu conversational dictionary]. Kinkōdō Shoseki, Tokyo.

Hideo Kirikae. 2003. *Ainu shin-yōshū jiten: tekisuto, bumpō kaisetsu tsuki* [Lexicon to Yukie Chiri's Ainu shin-yōshū with text and grammatical notes], Daigaku Shorin, Tokyo.

Yoshio Momouchi, Yasunori Azumi, and Yukio Kadoya. 2008. Research Note: Construction and Utilization of Electronic Data for Ainu Shin-yōsyū. *Bulletin of the Faculty of Engineering at Hokkai Gakuen University*, Vol. 35, pp. 159–171.

Hiroshi Nakagawa. 1995. *Ainugo Chitose Hōgen Jiten* [Dictionary of the Chitose dialect of Ainu]. Sōfūkan, Tokyo.

Karol Nowakowski, Michal Ptaszynski, and Fumito Masui. 2017. Improving Tokenization, Transcription Normalization and Part-of-speech Tagging of Ainu Language through Merging Multiple Dictionaries. In: *Proceedings of the 8th Language & Technology Conference (LTC'17)*, pp. 317-321.

Karol Nowakowski, Michal Ptaszynski, and Fumito Masui. 2018. A proposal for a unified corpus of the Ainu language. *IPSJ SIG Technical Report*, Vol. 2018-NL-237, pp. 1-6.

Michal Ptaszynski and Yoshio Momouchi. 2012. Part-of-Speech Tagger for Ainu Language Based on Higher Order Hidden Markov Model. *Expert Systems With Applications*, Vol. 39, Issue 14 (2012), pp. 11576-11582.

Kirsten Refsing. 1986. *The Ainu language. The morphology and syntax of the Shizunai dialect.* Aarhus University Press, Aarhus.

Masayoshi Shibatani. 1990. *The languages of Japan*. London: Cambridge University Press.

Suzuko Tamura. 1996. *Ainugo jiten: Saru hōgen. The Ainu-Japanese Dictionary: Saru dialect*. Sōfūkan, Tokyo.

A Feature set used in experiments with the SVMTool

Tables 9-12 present the feature sets defined for each of the four variants (Model 0/1/2/4) of the tagging model created in this research. Each of the tagging strategies offered by the SVMTool utilizes different variant(s) of the tagging model. For details, please refer to Giménez and Márquez (2012).

Feature category		Definition
Word features		w_{-2}, w_{-1}, w_0, w_1, w_2
POS features		p_{-2}, p_{-1}
Ambiguity classes		a_0, a_1, a_2
Maybe's		m_0, m_1, m_2
Word bigrams		(w_{-2}, w_{-1}), (w_{-1}, w_0), (w_0, w_1), (w_{-1}, w_1), (w_1, w_2)
POS bigrams		(p_{-2}, p_{-1}), (p_{-1}, p_1), (p_1, p_2)
Word trigrams		(w_{-2}, w_{-1}, w_0), (w_{-2}, w_{-1}, w_1), (w_{-1}, w_0, w_1), (w_{-1}, w_1, w_2), (w_0, w_1, w_2)
POS trigrams		(p_{-2}, p_{-1}, p_1), (p_{-1}, p_1, p_2)
Only for OoV words	Single characters	ca(1), cz(1)
	Prefixes	a(2), a(3), a(4)
	Suffixes	z(2), z(3), z(4)
	Lexicalized features	L (word length), SA (initial upper case), AA (all upper case), SN (starts with number), CA (any capital letter), CAA (several capital letters), CP (contains a period), CC (contains a comma), CN (contains a number), MW (contains a hyphen)

Table 9: Feature definition for Model 0.

Feature category		Definition
Word features		w_{-2}, w_{-1}, w_0, w_1, w_2
POS features		p_{-2}, p_{-1}, p_1, p_2
Ambiguity classes		a_0, a_1, a_2
Maybe's		m_0, m_1, m_2
Word bigrams		(w_{-2}, w_{-1}), (w_{-1}, w_0), (w_0, w_1), (w_{-1}, w_1), (w_1, w_2)
POS bigrams		(p_{-2}, p_{-1}), (p_{-1}, p_0), (p_{-1}, p_1), (p_0, p_1), (p_1, p_2)
Word trigrams		(w_{-2}, w_{-1}, w_0), (w_{-2}, w_{-1}, w_1), (w_{-1}, w_0, w_1), (w_{-1}, w_1, w_2), (w_0, w_1, w_2)
POS trigrams		(p_{-2}, p_{-1}, p_0), (p_{-2}, p_{-1}, p_1), (p_{-1}, p_0, p_1), (p_{-1}, p_1, p_2)
Only for OoVs	Prefixes	a(1), a(2), a(3), a(4)
	Suffixes	z(1), z(2), z(3), z(4)
	Lexicalized features	L, SA, AA, SN, CA, CAA, CP, CC, CN, MW

Table 10: Feature definition for Model 1.

Feature category		Definition
Word features		w_{-2}, w_{-1}, w_0, w_1, w_2
POS features		p_{-2}, p_{-1}
Ambiguity classes		a_0
Maybe's		m_0
Word bigrams		(w_{-2}, w_{-1}), (w_{-1}, w_0), (w_0, w_1), (w_{-1}, w_1), (w_1, w_2)
POS bigrams		(p_{-2}, p_{-1})
Word trigrams		(w_{-2}, w_{-1}, w_0), (w_{-2}, w_{-1}, w_1), (w_{-1}, w_0, w_1), (w_{-1}, w_1, w_2), (w_0, w_1, w_2)
Only for OoVs	Prefixes	a(1), a(2), a(3), a(4)
	Suffixes	z(1), z(2), z(3), z(4)
	Lexicalized features	L, SA, AA, SN, CA, CAA, CP, CC, CN, MW

Table 11: Feature definition for Model 2.

Feature category		Definition
Word features		w_{-2}, w_{-1}, w_0, w_1, w_2
POS features		p_{-2}, p_{-1}
Ambiguity classes		a_0, a_1, a_2
Maybe's		m_0, m_1, m_2
Word bigrams		(w_{-2}, w_{-1}), (w_{-1}, w_0), (w_0, w_1), (w_{-1}, w_1), (w_1, w_2)
POS bigrams		(p_{-2}, p_{-1}), (p_{-1}, p_1), (p_1, p_2)
Word trigrams		(w_{-2}, w_{-1}, w_0), (w_{-2}, w_{-1}, w_1), (w_{-1}, w_0, w_1), (w_{-1}, w_1, w_2), (w_0, w_1, w_2)
POS trigrams		(p_{-2}, p_{-1}, p_1), (p_{-1}, p_1, p_2)
Only for OoVs	Prefixes	a(1), a(2), a(3), a(4)
	Suffixes	z(1), z(2), z(3), z(4)
	Lexicalized features	L, SA, AA, SN, CA, CAA, CP, CC, CN, MW

Table 12: Feature definition for Model 4.