Resource-light acquisition of inflectional paradigms

Abstract

This paper presents a resource-light acquisition of morphological paradigms and lexicons for fusional languages. It builds upon Paramor (Monson, 2009), an unsupervised system, by extending it: (1) to accept a small seed of manually provided word inflections with marked morpheme boundary; (2) to handle basic allomorphic changes acquiring the rules from the seed. The algorithm has been tested on Czech, Slovene, German and Catalan and has shown increased F-measure in comparison with the Paramor baseline.

1 Introduction

Modern morphological analysers based on supervised machine learning and/or hand-written rules achieve very high accuracy. However, the standard way to create them for a particular language requires substantial amount of time, money and linguistic expertise. For example, the Czech analyzer by (Hajič, 2004) uses a manually created lexicon with 300,000+ entries. As a result, most of the world languages and dialects have no realistic prospect for morphological analyzers created in this way.

Various techniques have been suggested to overcome this problem, including unsupervised methods acquiring morphological information from an unannotated corpus. While completely unsupervised systems (e.g., (Goldsmith, 2001)) are scientifically interesting, shedding light on areas such as child language acquisition or general learneability, for many practical applications their precision is still too low. They also completely ignore linguistic knowledge accumulated over several millennia, often failing to discover rules that can be found in basic grammar books.

Lightly-supervised systems aim to improve upon the accuracy of unsupervised system by using a limited amount of resources. One of such systems for fusional languages is described in the paper.

Using a reference grammar, it is relatively easy to provide information about inflectional endings, possibly organized into paradigms. In some languages, an analyzer built on such information would have an acceptable accuracy (e.g., in English, most words ending in *ed* are past/passive verbs, and most words ending in *est* are superlative adjectives). However, in many languages, the number of homonymous endings is simply too high for such system to be useful. For example, the ending *a* has about 19 different meanings in Czech (Feldman and Hana, 2010).

Thus our goal is to discover inflectional paradigms each with a list of words declining according to it, in other words we discover a list of paradigms and a lexicon. But we do not attempt to assign morphological categories to any of the forms. For example, given an English corpus the program should discover that *talk, talks, talking, talked* are the forms of the same word, and that *work, push, pull, miss, . . .* decline according to the same pattern. However, it will not label *talked* as a past tense and not even as a verb.

This kind of shallow morphological analysis has applications in information retrieval (IR), for example search engines. For the most of the queries, users aren't interested only in particular word forms they entered but also in their inflected forms. In highly inflectional languages, such as Czech, dealing with morphology in IR is a necessity. Moreover, it can also be used as a basis for a standard morphological analyzer after labeling endings with morphological tags and adding information about closed-class/irregular words.

As the basis of our system, we chose Paramor (Monson, 2009), an algorithm for unsupervised induction of inflection paradigms and morphemic segmentation. We extended it to handle basic phonological/graphemic alternations and to accept seeding paradigm-lexicon information.

The rest of this paper is organized as follows: First, we discuss related work on unsupervised and semi-supervised learning. Follows a section about baseline Paramor model. After that, we motivate and describe our extension to it. Finally, we report results of experiments on Czech and Slovene.

2 Paramor

Our approach builds upon Paramor (Monson, 2009), an unsupervised approach for discovery of inflectional paradigms.

Due to data sparsity, not all inflections of a word are found in a corpus. Therefore Paramor does not require full paradigms, but instead works with partial paradigms, called *schemes*. A scheme contains a set of c(andidate)-suffixes and a set of c(andidate)-stems inflecting according to this scheme. The corpus must contain the concatenation of every c-stem with every c-suffix in the same scheme. Several schemes might correspond to a single morphological paradigm, because different stems belonging to the paradigm occur in the corpus in different set of inflections. The schemes are acquired in several steps:

1. Initialization: It first considers all possible segmentations of forms into candidate stems and endings.

2. Bottom-up Search: It creates schemes by joining endings that share a large number of associated stems.

3. Scheme clustering: Similar schemes (as measured by cosine similarity) are merged.

4. Pruning: Schemes proposing frequent morpheme boundaries not consistent with boundaries proposed by a character entropy measure are discarded.

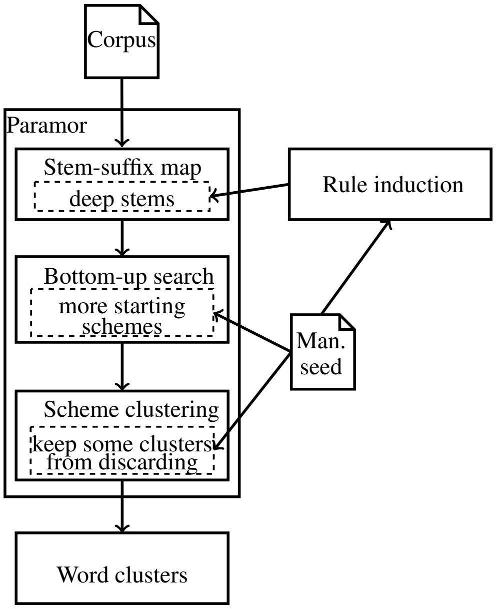

Figure 1: Altered Paramor's pipeline

Paramor works with types and not tokens. Thus it is not using any information about the frequency or context of forms.

3 Our Approach

3.1 Overview

We have modified the individual stems in Paramor's pipeline in order to use (1) a manually provided seed of inflected words divided into stems and suffixes; and (2) to take into account basic allomorphy of stems. Figure 1 shows phases of Paramor on the left with dashed boxes representing our alterations.

In the bottom-up search phase and the scheme cluster filtering phase, we use manually provided examples of valid suffixes and their grouping to subparadigms to steer Paramor towards creating more adequate schemes and scheme clusters. The data may also contain allomorphic stems, which we use to induce simple stem rewrite rules. Using these rules, some of the allomorphic stems in the corpus can be discovered and used to find more complete schemes.

3.2 Scheme seeding

The manual seed contains a simple list of inflected words with marked morpheme boundary. A simple example in English would be:

> *talk+0, talk+s, talk+ed, talk+ing*
> *stop+0, stop+s, stopp+ed, stopp+ing*
> *chat+0, chat+s, chatt+ed, chatt+ing*

The data are used to enhance Paramor's accuracy in discovering the correct schemes and scheme clusters in the following way:

1. In the bottom-up search, Paramor starts with single-affix schemes. We added a 2-affix scheme to the starting scheme set for every suffix pair form the manual data belonging to the same inflection. Note that we cannot simply add a scheme containing all the suffixes of the whole paradigm as many of the forms will not be present in the corpus.

2. Scheme clusters containing suffixes similar to some of the manually entered suffix sets are protected from the second phase of the cluster pruning. More precisely, a cluster is protected if at least half of its schemes share at least two suffixes with a particular manual suffix set.

3.3 Allomorphy

Many morphemes have several contextually dependent realizations, so-called allomorphs due to phonological/graphemic changes or irregularities. For example, consider the declension of the Czech word *matka* 'mother' in Table 1. It exhibits stem-final consonant change (palatalisation of *k* to č) triggered by the dative and local singular ending, and epenthesis (insertion of *-e-*) in the bare stem genitive plural.

Paramor ignores allomorphy completely (and so do most other unsupervised systems). There are at least two reasons to handle allomorphy. First, linguistically, it makes more sense to analyze *winning* as *win+ing* than as *winn+ing* or *win+ning*. For many applications, such as information retrieval, it is helpful to know that two morphs are variants of the same morpheme. Second, ignoring allomorphy makes the data appear more complicated and noisier than they actually are. Thus, the process of learning morpheme boundaries or paradigms is harder and less successful.

Case	Singular	Plural
nom	mat**k**+a	mat**k**+y
gen	mat**k**+y	mat**ek**+0
dat	mat**c**+e	mat**k**+ám
acc	mat**k**+u	mat**k**+y
voc	mat**k**+o	mat**k**+y
loc	mat**c**+e	mat**k**+ách
inst	mat**k**+ou	mat**k**+ami

Table 1: Declension of the word *matka* "mother". Changing part of the stem is in bold.

This latter problem might manifests itself in Paramor's bottom-up search phase: a linguistically correct suffix triggering a stem change might be discarded, because Paramor would not consider stem allomorphs to be variants of the same stem and c-stem ratio may drop significantly. Further more, incorrect c-suffixes may be selected.

However, for most languages the full specification of rules constraining allomorphy is not available, or at least is not precise enough. Therefore, we automatically induce a limited number of simple rules from the seed examples and/or from the scheme clusters obtained from the previous run of algorithm. Such rules both over and undergenerate, but nevertheless they do improve the accuracy of the whole system. For languages, where formally specified allomorphic rules are available, they can be used directly along the lines of (Tepper and Xia, 2010; Tepper and Xia, 2008). For now, we consider only stem final changes, namely vowel epenthesis (e.g., *matk-a – matek-0*) and alternation of the final consonant (e.g., *matk-a – matc-e*). The extension to other other processes such as root vowel change (e.g., English *foot – feet*) is quite straightforward, but we leave with for future work.

3.4 Stem change rule induction and application

Formally, the process can be described as follows. From every pair of stem allomorphs in the seed, $a\delta_1, a\delta_2$, where a is their longest common initial substring, with suffix sets F_1, F_2 we generate a correspondence rule $*\delta_1 \leftrightarrow *\delta_2$ / (F_1, F_2). A rule $*\delta_1 \leftrightarrow *\delta_2$ / (F_1, F_2) is applicable on an unordered pair of c-stems $x\delta_1$, $x\delta_2$ present in the corpus if:

1. C-suffix set of the c-stem $x\delta_1$ contains at least one of the suffixes from F_1 and contains no suffix from F_2.

2. C-suffix set of the c-stem $x\delta_2$ contains at least one of the suffixes from F_2 and contains no suffix from F_1.

These rules are used to generate underlying form of c-stems, which we call *deep* stems. We define relation $\leftrightarrow$ between two c-stems: $s_1 \leftrightarrow s_2$ iff there is a correspondence between s_1 and s_2 licensed by any rule. The deep stems then correspond to equivalence classes induced by the reflexive and transitive closure of $\leftrightarrow$. Bottom-up search and all the following phases of Paramor were modified to use the deep stems instead of the surface ones.

4 Experiments and results

We tested our approach on Czech, Slovene, German and Catalan lemmatised corpora, summarised in Table 2. The manual seed consisted of inflections of several lemmas for each languages (nouns, adjectives and verbs) obtained from a basic grammar overview. For example, listing inflections of 60 lemmas for Catalan took about 45 minutes. For Czech we also added information about the only two inflectional prefixes (negative prefix *ne* and superlative prefix *nej*). The decision which prefixes to consider inflectional and which not is to a certain degree an arbitrary decision (e.g., it can be argued that *ne* is a clitic and not a prefix), therefore it makes sense to provide such information manually.

Lang.	**Source**	**T**	**L**	**T $\geq$ 4**	**L $\geq$ 4**
cz	PDT1[2]	27k	13k	25k	12k
si	jos100k[3]	27k	15.5k	25k	14k
de	TIGER[4]	22k	17k	21k	16k
ca	Clic-TALP	11k	8k	10k	7k

Table 2: Corpora used in the evaluation. T – types, L – lemmas, T $\geq$ 4 – types longer than 4 characters

4.1 Evaluation method

We evaluated the experiments only on types at least 4 characters long to avoid most of the closed-class and irregular words. We used a pair-wise evaluation method similar to the one used in (Snover et al., 2002). For each pair of words we check whether they share a lemma and whether they share a common scheme cluster and we compute precision, recall and the standard balanced F-score.

4.2 Results

Results of the experiments are presented in Table 3. We used the following experiment settings: *no seed* – the baseline, Paramor was run without any seeding; *seed* – seed was used; *seed + pref.* – seed was used together with additional rules for two Czech inflectional prefixes.

Corpus	**Experiment**	**Seed**	**P**	**R**	**F1**
cz	no seed	0	87.65	57.75	69.63
	seed	18	87.14	62.43	72.74
	seed + pref.	18	86.31	71.59	**78.26**
si	no seed	0	69.98	80.40	74.83
	seed	9	69.60	82.74	**75.61**
de	no seed	0	56.02	64.87	60.12
	seed	11	54.67	72.69	63.40
ca	no seed	0	57.71	68.72	62.74
	seed	60	60.84	71.99	**65.95**

Table 3: Results for the for various corpora (Seed refers to the number of lemmas in the seed).

As can be seen from the results, the extra manual information indeed does help the accuracy of clustering words belonging to the same paradigms. F-measure and recall increase, precision decreases, but only slightly. What is not shown by the numbers is that in all languages more of the morpheme boundaries make linguistic sense because basic stem allomorphy is accounted for.

5 Conclusion

We have shown that providing a very little of easily obtainable information can improve the result of a purely unsupervised system. In the near future, we are planning to model a wider range of allomorphic alternations, try larger (but still easy to obtain) seeds and finally test the results on more languages.

[2] `http://ufal.mff.cuni.cz/pdt/`

[3] `http://nl.ijs.si/jos/jos100k-en.html`

[4] `http://www.ims.uni-stuttgart.de/projekte/TIGER`

References

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL '02, pages 21–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pages 106–113. Finland: Espoo.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4:3:1–3:34, February.

Anna Feldman and Jirka Hana. 2010. *A resource-light approach to morpho-syntactic tagging*. Rodopi, Amsterdam/New York, NY.

John A. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Jan Hajič. 2000. Morphological Tagging: Data vs. Dictionaries. In *Proceedings of ANLP-NAACL Conference*, pages 94–101, Seattle, Washington, USA.

Jan Hajič. 2004. *Disambiguation of Rich Inflection: Computational Morphology of Czech*. Karolinum, Charles University Press, Praha.

Jiri Hana, Anna Feldman, and Chris Brew. 2004. A resource-light approach to Russian morphology: Tagging Russian using Czech resources. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 222–229, Barcelona, Spain, July. Association for Computational Linguistics.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, pages 78–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Monson. 2009. *ParaMor: From Paradigm Structure to Natural Language Morphology Induction*. Ph.D. thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

Kemal Oflazer, Sergei Nirenburg, and Marjorie McShane. 2001. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational Linguistics*, 27(1):59–85.

Jorma Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co, Singapore.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the North American chapter of the Association for Computational Linguistics*, pages 183–191.

Matthew G. Snover, Gaja E. Jarosz, and Michael R. Brent. 2002. Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In *In Proc. ACL Worksh. Morphol. & Phonol. Learn*, pages 11–20.

Michael Tepper and Fei Xia. 2008. A hybrid approach to the induction of underlying morphology. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008), Hyderabad, India, Jan 7-12*, pages 17–24.

Michael Tepper and Fei Xia. 2010. Inducing morphemes using light knowledge. *ACM Trans. Asian Lang. Inf. Process.*, 9:3:1–3:38, March.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics*, pages 207–216.