Unsupervised Learning of A-Morphous Inflection with Graph Clustering

Maciej Janicki
University of Leipzig, Master's Programme in Computer Science
macjan@o2.pl

Abstract

This paper presents a new approach to unsupervised learning of inflection. The problem is defined as two clusterings of the input wordlist: into lexemes and into forms. Word-Based Morphology is used to describe inflectional relations between words, which are discovered using string edit distance. A graph of morphological relations is built and clustering algorithms are used to identify lexemes. Paradigms, understood as sets of word formation rules, are extracted from lexemes and words belonging to similar paradigms are assumed to have the same inflectional form. Evaluation was performed for German, Polish and Turkish and the results were compared to conventional morphological analyzers.

1 Introduction

Inflection is the part of morphology concerned with systematic variation of word forms in different syntactic contexts. Because this variation is expressed with patterns that appear over many words, it can be discovered using as little data as a plain wordlist. In the following sections, I will present a general, language-independent approach for the unsupervised learning of inflection, which makes use of simple algorithms, like string edit distance and graph clustering, along with Word-Based Morphology, a morphological theory that rejects the notion of morpheme.

It seems plausible to distinguish inflection from other morphological phenomena (derivation, compounding). Inflection examines the correspondence between items of the lexicon and their surface realizations, while derivation and compounding operate inside lexicon (Stump, 1998). The purpose of morphological annotation of texts, for example for Information Retrieval or Part-of-Speech tagging, is mostly to determine, for a given word, which lexical item (*lexeme*) in which syntactic context (*form*) it realizes. We are less interested in how this lexical item was created. For example, we would like to know that the words *gives* and *give* express the same meaning, while *giver* and *forgive* mean something different. Therefore, what we need is an *inflectional*, rather than a full *morphological* analysis. In the task of unsupervised learning of morphology, distinguishing inflection from derivation is a major challenge. Despite its usefulness, it has not been approached in state-of-the-art systems.

2 Related Work

The task of unsupervised learning of morphology has an over fifty years long history, which is exhaustively presented by Hammarström and Borin (2011). The most popular formulation of this problem is learning segmentation of words into morphemes. State-of-the-art systems for learning morpheme segmentation include Morfessor (Creutz et al., 2005) and Linguistica (Goldsmith, 2006). Both rely on optimization-based learning techniques, such as Minimum Description Length, or Maximum A-Posteriori estimate.

Some other authors use the approach that is called *group and abstract* by Hammarström and Borin (2011). First, they group the words according to some similarity measure, which is supposed to give high values for morphologically related words. Then, they abstract morphological rules from the obtained groups. Yarowsky and Wicentowski (2000) use a combination of four different similarity functions: string edit distance, contextual similarity, frequency similarity and transformation probabilities. Kirschenbaum et al. (2012) use contextual similarity.

The learning of morphology has already been formulated as a clustering problem by Janicki

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 93–99,
Hissar, Bulgaria, 9-11 September 2013.

(2012), which uses mutual information to identify inflectional paradigms, a method that is also employed here. However, the algorithm presented there handles only suffix morphologies and only clustering into lexemes is performed. Word-based morphology has been previously used for the unsupervised learning task by Neuvel and Fulop (2002), but for the purpose of generating unseen words, rather than inducing inflectional analysis.

3 Morphology without Morphemes

Traditional morphological theory uses the notion of *morpheme*: the smallest meaningful part of a word. Morphological analysis of a word is typically understood as splitting it into morphemes and labeling each morpheme with semantic or functional information. However, morphological operations often include phenomena that are not plausibly described by the notion of morpheme. That is why alternative theories were proposed, in which variations between word forms are described with rules operating on phonological representations of whole words, without isolating morphemes or setting boundaries.

The term *Word-Based Morphology* can be traced back to Aronoff (1976). In his analysis of derivational morphology, he shows that the minimal meaningful elements of a language are words, rather than morphemes. He formulates the hypothesis that new words are formed by *Word-Formation Rules*, which always operate on a whole existing word.

Aronoff's ideas motivate the theory developed by Anderson (1992), which presents a complete, a-morphous description of morphology, while maintaining the distinction between inflection, derivation and compounding. In Anderson's theory, the lexicon is a set of *lexical stems*, where lexical stem is defined as "word minus its inflectional material". Turning stems to surface words is done by word-formation rules, which are triggered by particular syntactic contexts. The inflectional system of the language is the set of word-formation rules, along with their applicability conditions. Derivation is performed by word-formation rules of a different type, which operate on stems to form other stems, rather than surface words.

Finally, Ford et al. (1997) present an entirely word-based morphological theory, which radically criticizes all kinds of abstract notions of morphological analysis (like stem or morpheme). It claims that there is only one kind of rule, which is used to describe systematic patterns between surface words, and which can be represented in the following form:

$$/\mathrm{X}/_{\alpha} \leftrightarrow /\mathrm{X}'/_{\beta}$$

where X and X$'$ are words and α and β morphological categories. No distinction is made between inflection and derivation.

Since in the task of unsupervised learning of inflection the only available data are surface words, the last theory seems especially plausible. A candidate morphological rule can be extracted from virtually any pair of words. The "real" rules can be distinguished from pairs of unrelated words basing on their frequency and co-occurrence or interaction with other rules. However, the application of Ford et al.'s theory will here be restricted to inflection, with the purpose of finding *lexemes* – clusters of words connected by inflectional rules. The lexemes provide enough information to derive stems and word-formation rules in the sense of Anderson's theory, which can be further used for learning derivation and compounding, since, in my opinion, the latter are better described as relations between lexemes, rather than surface words.

4 What to Learn?

Conventional inflectional analyzers, like for example Morfeusz[1] (Woliński, 2006) for Polish, provide two pieces of information for each word: the *lexeme*, to which this word belongs, and the *tag*, describing the inflectional form of it. For example, for the German[2] word *Häusern* (dative plural of the noun *Haus* 'house'), the correct analysis consists of the lexeme HAUS and the tag 'Dative Plural'.

Our task is to train an analyzer, which will provide similar analysis, using only a plain list of words. We certainly cannot achieve exactly the same, because we do not have access to lemmas and labels for grammatical forms. However, we can identify a lexeme by listing all words that belong to it, like HAUS $=$ $\{$*Haus*, *Hauses*, *Häuser*, *Häusern*$\}$. Similarly, we will identify an inflectional form by listing all

[1] See `http://sgjp.pl/demo/morfeusz` for an online demo.

[2] German is used as source of examples, because English inflection is often too simple to illustrate the discussed issues.

words that have this form. For example, the German 'Dative Plural' will be defined as: DAT.PL $=$ $\{$*Bäumen*, *Feldern*, *Häusern*, *Menschen*, $\ldots\}$.

In this way, inflectional analysis can be seen as two clustering problems: grouping words into *lexemes* and into *forms*. If an unsupervised analyzer is able to produce those two clusterings, then the results could be converted into a 'proper' inflectional dictionary with a minimal human effort: annotating each cluster with a label (lemma or stem for lexemes and inflectional tag for forms) which cannot be extracted automatically.

In my opinion, formulating inflectional analysis as a clustering problem has certain advantages over, for instance, learning morpheme segmentation. The clustering approach provides similar information as conventional inflectional analyzers, and can be directly used in many typical applications, like lexeme assignment (equivalent to stemming/lemmatization) in Information Retrieval, or grammatical form labeling, for example for the purpose of Part-of-Speech Tagging. It also gets rid of the notions of morpheme and segmentation, which depend on the morphological theory used, and can be problematic.[3] Finally, well-established clustering evaluation measures can be used for evaluation.

5 The Algorithm

5.1 Building the Morphology Graph

At first, for each word in the data, we find similar words wrt. string edit distance. An optimized algorithm, similar to the one presented by Bocek et al. (2007), is used to quickly find pairs of similar words. For each word, we generate substrings through deletions. We restrict the number of substrings by restricting the number of deletions to five characters at the beginning of the word, five at the end and five in a single slot inside the word, whereas the total number must not exceed half of the word's length. This is enough to capture almost all inflectional operations. Then, we sort the substrings and words that share a substring are considered similar.

The systematic variation between similar words is described in terms of Word-Based Morphology: for each pair (w_1, w_2), we extract the operation needed to turn w_1 into w_2. We formulate it in terms of adding/substracting a prefix, performing a certain internal modification (insertion, substitution, deletion) and adding/substracting a suffix. For example, the operation extracted from the pair (*senden*, *gesandt*) would be *-:ge-/e:a/-en:-t*, while the operation extracted from the pair (*absagen*, *sagten*) would be *ab-:-/-:t/-:-* (substract the prefix *ab-*, insert *-t-*, no suffixes).

I believe that the notion of *operation*, understood as in the above definition, is general enough to cover almost all inflectional phenomena and does not have a bias towards a specific type of inflection. In particular, prefixes are treated exactly the same way as suffixes. The locus and context of internal modification is not recorded, so the pairs *sing:sang*, *drink:drank* and *begin:began* are described with the same operation. This is important, because the algorithm involves computing frequency of the operations. Note that this also means that operations cannot be used for deriving one word from another unambigously, but this is not needed in the algorithm presented here.

From the above data, we build the *morphology graph*, in which the vertices are the words, and the edges are operations between words. Because every operation is reversible, the graph is undirected. We assign a weight to every edge, which is the natural logarithm of the frequency of the corresponding operation: frequent operations are likely to be inflectional rules, while the infrequent are mostly random similarities between words. We set a minimal frequency needed to include an operation in the graph on 1/2000 of the size of the input wordlist.

5.2 Clustering Edges

Inflectional rules tend to occur in groups, called *paradigms*. For example, if a German noun uses the *-er* suffix to form nominative plural, it also uses *-ern* for dative plural and probably *-es* for genitive singular. This property can be expressed by means of mutual information, which has been described by Janicki (2012): inflectional rules that belong to the same paradigm tend to have high mutual information values, measured over the probability of occurring with a random word.

The morphology graph stores for each word the information, which operations can be applied to it. These operations can be inflectional rules, as well as derivational rules and random similarities. By clustering the operations according to mutual in-

[3] See for example the discussion of evaluation problems in (Goldsmith, 2006).

formation, we identify groups of operations which show strong interdependence, which means that they are likely to be paradigms or fragments of those. Derivational rules and random similarities show mostly no interdependence, so they form singleton clusters.

We use the complete-linkage clustering with a fixed threshold value. It is much faster than the hierarchical clustering applied by Janicki (2012) and produces similar results. The threshold value does not have much influence on the final results: it should not be too high, so that real paradigms are split. If it is too low and some non-inflectional operations are mixed together with inflectional paradigms, it can still be fixed in the next step. I used the threshold value 0.001 in all my experiments and it performed well, regardless of language and corpus.

5.3 Clustering the Graph into Lexemes

The previous steps provide already some clues about which words can belong to the same lexeme. Operations are assigned weights according to their frequency, and interdependent operations are grouped together. Now we can apply a graph clustering algorithm, which will split our graph into lexemes, using the above information.

We use the Chinese Whispers clustering algorithm (Biemann, 2006). It is very simple and efficient and it does not require any thresholds or parameters. At the beginning, it assigns a different label to every vertex. Then it iterates over the vertices in random order and each vertex receives labels passed from its neighbours, from which it chooses the most "promoted" label, according to the sum of weights of the edges, through which this label is passed. The procedure is repeated as long as anything changes. The algorithm has already been succesfully used for many NLP problems, but, to my knowledge, not for unsupervised learning of morphology.

A slight modification is made to the Chinese Whispers algorithm to take advantage of the edge clustering performed in the previous step. Every word is split into multiple vertices: one for each edges cluster. During the clustering, they are treated as completely different vertices. It ensures us that we will not pick non-inflectional operations together with inflectional ones or merge two distinct paradigms. After the clustering however, we again leave only one vertex per word: the one whose label has the biggest score understood the same way as in Chinese Whispers algorithm. Finally, by grouping words together according to their label, we obtain the clustering into lexemes.

5.4 Extracting Paradigms and Forms

Given the lexemes, we can easily compute the *paradigm* for each word, understood as the set of operations that generates this word's whole lexeme. Paradigms will be used to derive clustering into forms. We observe that if two words have the same paradigm, they almost certainly share the grammatical form, which is illustrated in table 1. Unfortunately, the reverse is not true: words that share the form do not necessarily share the paradigm. Firstly, in every corpus there are many missing word forms. Continuing the example from table 1, let's assume that the words *Mannes* and *Bändern* are missing. Then, the words $\{$*Haus*, *Mann*, *Band*$\}$ would all have different, although similar, paradigms. The second reason is that one form may be created in different ways, depending on the inflection class of the lexeme. The operation *:/a:ä/:er* is only one of many ways of forming nominative plural in German.

A quick solution to the above problems is clustering paradigms according to cosine similarity. For each paradigm P, we define a corresponding vector of operation frequencies:

$$\vec{v}[op] = \begin{cases} ln(freq(op)) & \text{if } op \in P \\ 0 & \text{if } op \notin P \end{cases}$$

where op is a morphological operation and $freq(op)$ its number of occurences. The similarity between two paradigms is defined as 0 if they share less then a half of their operations, and as the cosine of the angle between their vectors otherwise. We use the Chinese Whispers algorithm again for clustering paradigms. Finally, we group the words into forms using the assumption that two words have the same form, if their paradigms belong to the same cluster.

6 Evaluation

For the evaluation of the clusterings, I used the extended BCubed measure (Amigó et al., 2009). Contrary to other popular clustering evaluation measures (e.g. cluster purity), it penalizes all possible kinds of errors and no cheat strategy exists for it. For example, it is sensitive to splitting a correct cluster into parts. It also allows overlapping clusters and classes, which can be the case in

Word	Form	Paradigm
Haus, Mann, Band	NOM.SG	:/:/:es, :/a:ä/:er, :/a:ä/:ern
Hauses, Mannes, Bandes	GEN.SG	:/:/es:, :/a:ä/s:r, :/a:ä/s:rn
Häuser, Männer, Bänder	NOM.PL	:/ä:a/er:, :/ä:a/r:s, :/:/:n
Häusern, Männern, Bändern	DAT.PL	:/ä:a/ern:, :/ä:a/rn:s, :/:/n:

Table 1: The correspondence between form and paradigm. Same paradigm implies same form.

inflectional analysis, as some surface words may be realizations of multiple lexemes. The results are given in the usual terms of Precision, Recall and F-measure.

I used corpora from Leipzig Corpora Collection[4] to build input wordlists of approximately 200,000 words for German, Polish and Turkish. The golden standard clusterings were constructed by analyzing the input data with conventional morphological analyzers: Morfeusz (Woliński, 2006) for Polish, Morphisto (Zielinski and Simon, 2009) for German and TRmorph (Çöltekin, 2010) for Turkish. Words that have the same lemma, according to the morphological analyzer used, were grouped into golden standard lexemes, and words that share all their inflectional tags – into golden standard form clusters. Words that were unknown to the analyzer, were not included in results calculation.

The evaluation results for lexeme clustering are given in table 2. All datasets achieve good precision, around 90 %. The recall for Polish and German is also high. In addition to performing well on suffix-based inflectional operations, the algorithm also succeeded in finding many German plural forms that involve vowel alternation (umlaut). Problematic is the significantly lower recall score for Turkish. The reason is the Turkish agglutinative morphology, with very large paradigms, especially for verbs. Complex forms are often treated as derivations and large verb lexemes are split into parts.

Testing set	Precision	Recall	F-measure
German	87.8 %	79.8 %	83.5 %
Polish	89.0 %	80.1 %	84.3 %
Turkish	92.9 %	41.4 %	57.3 %

Table 2: Lexeme evaluation.

Testing set	Precision	Recall	F-measure
German	64.1 %	12.8 %	21.4 %
Polish	61.5 %	34.8 %	44.4 %
Turkish	45.6 %	10.8 %	17.5 %

Table 3: Form evaluation.

In general, the algorithm performs well in distinguishing inflection from derivation, as long as lexemes have enough inflected forms. The Chinese Whispers algorithm identifies strongly interconnected sets and inflection usually involves more forms and more frequent operations than derivation. A problem emerges for rare lexemes, which are only represented by one or two words in the corpus, and which take part in many common derivations, like the German prefixing. It can happen that derivational operations connect them stronger than inflectional ones, which results in clusterings according to derivational prefixes. For example, we obtain $\{$*abdrehen*, *aufdrehen*, ...$\}$ in one cluster and $\{$*abgedreht*, *aufgedreht*, ...$\}$ in another. This is one of the most common mistakes in the German dataset and it should be addressed in further work.

Table 3 shows the results for clustering into forms. They are considerably lower than in lexeme clustering. The main reason for low precision is that there are some distinctions in morphosyntactical information that are not visible in the surface form, like gender in German. The second reason are small paradigms that are induced for words, for which only a few forms appear in the corpus. Small paradigms do not provide enough grammatical information and lead to clustering distinct forms of rare words together. Recall scores are even lower than precision, which is caused by the issues discussed in section 5.4. Clustering paradigms according to cosine similarity is by far not enough to solve these problems.

Comparing my algorithm to other authors' work is difficult, because, to my knowledge, no other approach is designed for the definition of the problem presented here – clustering words into lex-

[4] `http://corpora.uni-leipzig.de`

emes and forms. Comparing it to morpheme segmentation algorithms would need converting morpheme segmentation to lexemes and forms, which is not a trivial task.

7 Conclusion

I have shown that a full inflectional analysis can be defined as two clusterings of the input wordlist: into *lexemes* and *forms*. My opinion is that for the purpose of unsupervised learning of inflection, such output is more useful and easier to evaluate, than morpheme segmentation. From a theoretical view, my approach can be seen as a minimalist description of inflection, which uses only words as data and describes the desired information (lexeme and form) in terms of word sets, while getting rid of any abstract units of linguistic analysis, like morpheme or stem.

Further, I have provided an algorithm, which learns inflection through graph clustering, based on the Word-Based theory of morphology. I have compared it to the output of state-of-the-art handcrafted morphological analyzers. The algorithm performs especially well in the task of clustering into lexemes for inflectional morphologies and is capable of discovering non-concatenative operations. Many errors are due to missing word forms in the corpus. The output can be applied directly or used to minimize human effort while constructing an inflectional analyzer.

The presented algorithm will be subject to further work. The results of lexeme clustering could probably be improved with a more careful scoring of operations, rather than just simple frequency. Other possibly useful features should be examined, perhaps making use of the information available in unannotated corpora (like word frequencies or context similarity). A better algorithm for clustering into forms is also needed, because cosine similarity does not give satisfactory results. Finally, I will try to approach derivation and compounding with methods similar to the one presented here.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486.

Stephen R. Anderson. 1992. *A-Morphous morphology*. Cambridge University Press.

Mark Aronoff. 1976. *Word formation in generative grammar*. Linguistic inquiry. Monographs. MIT Press.

Chris Biemann. 2006. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80.

Thomas Bocek, Ela Hunt, and Burkhard Stiller. 2007. Fast Similarity Search in Large Dictionaries. Technical Report ifi-2007.02, Department of Informatics, University of Zurich, April. http://fastss.csg.uzh.ch/.

Çağrı Çöltekin. 2010. A Freely Available Morphological Analyzer for Turkish. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Mathias Creutz, Krista Lagus, and Sami Virpioja. 2005. Unsupervised morphology induction using morfessor. In Anssi Yli-Jyrä, Lauri Karttunen, and Juhani Karhumäki, editors, *Finite-State Methods and Natural Language Processing, 5th International Workshop*, volume 4002 of *Lecture Notes in Computer Science*, pages 300–301. Springer.

Alan Ford, Rajendra Singh, and Gita Martohardjono. 1997. *Pace Pāṇini: Towards a word-based theory of morphology*. American University Studies. Series XIII, Linguistics, Vol. 34. Peter Lang Publishing, Incorporated.

John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Nat. Lang. Eng.*, 12(4):353–371.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Maciej Janicki. 2012. A Lexeme-Clustering Algorithm for Unsupervised Learning of Morphology. In Johannes Schmidt, Thomas Riechert, and Sören Auer, editors, *SKIL 2012 - Dritte Studentenkonferenz Informatik Leipzig*, volume 34 of *Leipziger Beiträge zur Informatik*, pages 37–47. LIV, Leipzig.

Amit Kirschenbaum, Peter Wittenburg, and Gerhard Heyer. 2012. Unsupervised morphological analysis of small corpora: First experiments with kilivila. In Frank Seifart, Geoffrey Haig, Nikolaus P. Himmelmann, Dagmar Jung, Anna Margetts, and Paul Trilsbeek, editors, *Potentials of Language Documentation: Methods, Analyses, and Utilization. Language Documentation & Conservation Special Publication*, pages 25–31. Manoa: University of Hawaii Press.

Sylvain Neuvel and Sean A. Fulop. 2002. Unsupervised learning of morphology without morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL '02, pages 31–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gregory T. Stump. 1998. Inflection. In Andrew Spencer and Arnold M. Zwicky, editors, *The Handbook of Morphology*, pages 13–43. Blackwell Publishing.

Marcin Woliński. 2006. Morfeusz a Practical Tool for the Morphological Analysis of Polish. In Mieczysław A. Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent Information Processing and Web Mining*, Advances in Soft Computing, pages 503–512. Springer-Verlag, Berlin.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 207–216.

Andrea Zielinski and Christian Simon. 2009. Morphisto – an open source morphological analyzer for german. In *Proceedings of the 2009 conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008*, pages 224–231, Amsterdam, The Netherlands. IOS Press.