MorphoNet: Exploring the Use of Community Structure for Unsupervised Morpheme Analysis

Delphine Bernhard*

ILES group, LIMSI-CNRS, Orsay, France
delphine.bernhard@limsi.fr

Abstract. This paper investigates a novel approach to unsupervised morphology induction relying on community detection in networks. In a first step, morphological transformation rules are automatically acquired based on graphical similarities between words. These rules encode substring substitutions for transforming one word form into another. The transformation rules are then applied to the construction of a lexical network. The nodes of the network stand for words while edges represent transformation rules. In the next step, a clustering algorithm is applied to the network to detect families of morphologically related words. Finally, morpheme analyses are produced based on the transformation rules and the word families obtained after clustering. While still in its preliminary development stages, this method obtained encouraging results at Morpho Challenge 2009, which demonstrate the viability of the approach.

1 Introduction

Unsupervised morphology induction, which is the goal of the Morpho Challenge competition series [1], consists in automatically discovering a word's morphemes using only minimal resources such as a list of the words in the target language and a text corpus. Ideally, unsupervised algorithms should be able to learn the morphology of a large variety of languages; for Morpho Challenge 2009, the target languages were English, Finnish, German, Turkish and Arabic.

For our participation at Morpho Challenge 2009 we developed a novel method for unsupervised morphology induction called *MorphoNet*. MorphoNet relies on a *network* representation of morphological relations between words, where nodes correspond to whole word forms and edges encode morphological relatedness. Networks have been successfully used in recent years to represent linguistic phenomena for tasks such as word clustering [2], word sense disambiguation [3], summarisation, or keyword extraction [4]. Moreover, network-based methods have been shown to perform well for a wide range of NLP applications. In line with this body of research, we propose to represent morphological phenonema as a network. This approach has two major advantages. First, it is theoretically related to linguistic theories such as the Network Model by J. Bybee [5] or

* This work was done while the author was at the UKP Lab, Darmstadt, Germany.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 598–608, 2010.

whole word morphology [6]. It differs from traditional linear concatenative approaches to morphology in that words are not explicitely split into a sequence of morphemes, but related to one another through morphological transformations. Second, it enables the use of effective network-based clustering and ranking methods. Our model thus benefits from research done on graphs in other domains such as sociology [7] or other areas of NLP. We especially investigate the use of *community structure* for morphology induction. Networks with community structure contain groups of nodes with dense interconnections; in our case, communities correspond to families of morphologically related words. Communities can be automatically identified in networks with community detection algorithms. To our knowledge, this is the first time that community detection algorithms are applied to the task of unsupervised morphology induction.

Though in its very early development stages, the approach yields promising results at Morpho Challenge 2009 when compared to standard baselines such as the Morfessor algorithms [8, 9].

The article is structured as follows. In the next section, we report related work. Then, we describe our method for building lexical networks. In Sect. 4, we explain how word families can be discovered based on the network structure, while in Sect. 5 we detail our approach for obtaining morpheme analyses. Evaluation results are given in Sect. 6.

2 Related Work on Morphology Induction

Morphological analysis is useful for many applications like speech recognition and synthesis, machine translation or information retrieval. However, all these applications of morphology necessitate morphological resources which are not available for all languages, or, when available, are often incomplete. Much research has therefore been devoted to the unsupervised acquisition of morphological knowledge.

Methods for the unsupervised acquisition of morphological knowledge can be classified according to the intended result: (i) identification of morphologically related words (*clustering*), (ii) splitting of words into morphs (*segmentation*), and (iii) identification of morphemes (*analysis*). Morpheme analysis is the goal of the latest Morpho Challenge competitions, while for some applications, such as information retrieval, it is often sufficient to retrieve morphologically related words without proceeding to a full analysis. The identification of morphologically related words has been attempted by unsupervised methods [10] as well as approaches using dictionaries as input data [11].

Segmentation is certainly the method which has gathered the largest amount of interest in the NLP research community [8, 12–14]. It follows linear concatenative approaches to morphology such as item-and-arrangement, which postulates that words are formed by putting morphemes together. There are, however, some well known limitations to purely concatenative approaches, which are seldom dealt with by unsupervised segmentation methods. These limitations include ablaut, umlaut, and infixation. Contrarily to unsupervised morpheme segmentation methods, MorphoNet makes no assumption on the internal structure and morphotactics of words. It identifies flexible word transformation rules which encode substring substitutions for transforming one word form into another. These transformation rules are not limited to concatenative processes such as prefixation or suffixation (see Sect. 3.2) and thus aim at addressing some of the limitations of concatenative approaches.[1]

Unsupervised methods rely on many properties for morphology induction, which are too numerous to be listed here. The most obvious cue is usually *graphical relatedness*: two words which share a long enough common substring are likely to be morphologically related. Graphical relatedness can be estimated by measures of orthographic distance [15] or by finding the longest initial (or final) substring [16, 17]. Our system is related to these methods in that it uses fuzzy string similarity to bootstrap the morphology induction process.

3 Lexical Networks

3.1 Use of Graphs for Morphology Induction

A network can be mathematically represented as a graph. Formally, a graph G is a pair (V, E), where V is a set of vertices (nodes) and $E \subseteq V \times V$ is a set of edges (lines, links). The main advantage of graphs is that they make it possible to take into account multiple dependencies across elements, so that the whole network plays an important role on the results obtained for a single element.

The lexical networks built by our method consist of word nodes linked by edges which encode morphological relations. Similar lexical networks have been previously described by Hathout [18]. Our approach differs however from Hathout's in two main aspects: (i) it uses only a raw list of words as input, while Hathout's method acquires morphological links from WordNet, and (ii) we attempt to take a broader range of morphological phenomena into account by acquiring morphological transformation rules which are not limited to suffixation.

3.2 Acquisition of Morphological Transformation Rules

The first step in our method consists in acquiring a set of *morphological transformation rules*. Morphological transformation rules make it possible to transform one word into another by performing substring substitutions. We represent a rule R with the following notation: `pattern` → `repl`, where `pattern` is a regular expression and `repl` is the replacement with backreferences to capturing groups in the pattern. For instance, the rule `^(.+)ly$` → `\1` applies to the word *totally* to produce the word *total*.

Tranformation rules are in principle not limited to concatenative processes, which should be especially useful for languages such as Arabic, e.g. when inducing rules for word pairs such as *kataba* (he wrote) and *kutiba* (it was written).

[1] However, MorphoNet does not address cases of morphologically related words with no orthographic overlap, such as *be* and *was*.

These rules are acquired using a subset L of the wordlist W provided for each language to avoid noise given the substantial length of the word lists provided for MorphoChallenge. In our experiments, we used the 10,000 most frequent words whose length exceeds the average word (type) length.[2] The method used to acquire the rules is described in detail in Algorithm 1.

Algorithm 1. Procedure for the acquisition of morphological transformation rules, given an input list of words L

```
1: $rules \leftarrow \emptyset$
2: $n \leftarrow \mathsf{len}(L)$
3: **for** $i = 1$ to $n$ **do**
4:    $w \leftarrow L[i]$
5:    $matches \leftarrow \mathsf{get\_close\_matches}(w, L[i+1:n])$
6:    **for** $w_2$ in $matches$ **do**
7:       $r \leftarrow \mathsf{get\_rule\_from\_word\_pair}(w, w_2)$
8:       add $r$ to $rules$
9:    **end for**
10: **end for**
11: **return** $rules$
```

For each word w in the list L we retrieve graphically similar words (Line 5, get_close_matches) using a *gestalt* approach to fuzzy pattern matching based on the Ratcliff-Obershelp algorithm.[3] This string comparison method computes a measure of the similarity of two strings relying on the number of characters in matching subsequences. For example, given the target word *democratic*, the following close matches are obtained: *undemocratic*, *democratically*, *democrats*, *democrat's*, *anti-democratic*. We then obtain rules (Line 7, get_rule_from_word_pair) by comparing the target word with all its close matches and identifying the matching subsequences;[4] for instance given the word *democratic* and its close match *undemocratic*, we obtain the following rule: `^un(.+)$` → `\1`.

We have kept all rules which occur at least twice in the training data.[5] Moreover, no attempt is made to distinguish between inflection and derivation.

Table 1 lists the number of transformation rules obtained from the datasets provided for Morpho Challenge 2009[6] along with some examples:

3.3 Construction of a Lexical Network

Once transformation rules have been acquired, they are used to build a lexical network represented as a graph. Nodes in the graph represent words from the

[2] Except for Arabic, where there are only 9,641 word forms which are longer than the average word length in the vowelized version and 6,707 in the non-vowelized version.

[3] We used the implementation provided by the Python *difflib* module with the cutoff argument set to 0.8.

[4] Matching subsequences are identified by the get_matching_blocks Python method.

[5] For Arabic, we even kept all rules given the small size of the input word list.

[6] `http://www.cis.hut.fi/morphochallenge2009/`

Table 1. Morphological transformation rules acquired for the input datasets

Language	# rules	Examples	
English	834	`^re(.+)s$` → `\1`	`^(.+)'s$` → `\1`
Finnish	1,472	`^(.+)et$` → `\1ia`	`^(.+)ksi$` → `\1t`
German	771	`^(.+)ungen$` → `\1t`	`^(.+)ge(.+)t$` → `\1\2en`
Turkish	3,494	`^(.+)n(.+)$` → `\1\2`	`^(.+)nde$` → `\1`
Arabic vowelized	8,974	`^(.+)iy(.+)$` → `\1uw\2`	`^(.+)K$` → `\1N`
Arabic non-vowelized	2,174	`^(.+)wA$` → `\1`	`^(.+)hm$` → `\1`

input word list W. Two words w_1 and w_2 are connected by an edge if there exists a transformation rule R such that $\mathrm{R}(w_1) = w_2$. The graph obtained using this method is directed based on the direction of the rules applied. Figure 1 displays an example lexical network.

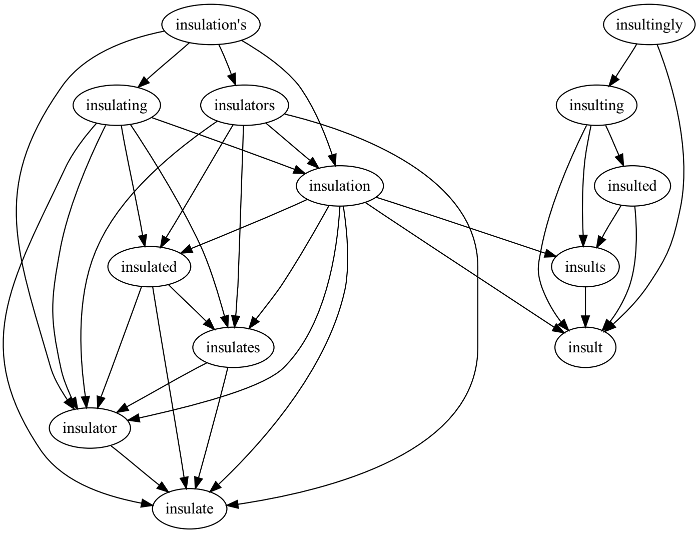

Fig. 1. Example lexical network

4 Acquisition of Word Families

The graphs we obtain usually contain one large connected component, along with smaller connected components. Extracting connected components is thus not reliable enough to identify word families, i.e. groups of words which are related both semantically and orthographically. For instance, the lexical network depicted in Fig. 1 contains one large connected component, which clearly consists of two different word families. The induction of word families can be formulated as a classical problem of community detection in graphs. Communities correspond to groups of tightly-knit nodes characterised by a high intra-group density and a lower inter-group density [19]. There are several methods to detect communities in graphs. Markov Clustering [20] for instance consists in partitioning a graph by simulating random walks in the graph; it has been used to detect communities in a graph of nouns by Dorow et al. [21]. The community detection method described by Newman [19] has been applied to natural language data by Matsuo et al. [2] for graphs based on word similarity measures by web counts.

The method proposed by Newman relies on the modularity function Q which measures the quality of a division of a graph into communities. The advantages of this method are that it is not necessary to know the number of communities beforehand and it needs no fine parameter tuning. Modularity compares the number of edges within communities to the number of expected edges:

$$Q = \sum_i (e_{ii} - (\sum_j e_{ij})^2) \tag{1}$$

where e_{ii} is the fraction of the edges in the network that connect nodes within community i, e_{ij} is one-half of the fraction of edges in the network that connect nodes in community i to those in community j and $\sum_j e_{ij}$ is the fraction of edges connected to nodes in community i.

A good division corresponds to more edges within communities than would be expected by random chance, that is a positive modularity value Q. Modularity is high when there are many edges within communities and few between them. By applying Newman's algorithm on the lexical network of Fig. 1, two communities are identified: {*insulation's, insulating, insulators, insulation, insulated, insulates, insulator, insulate*} and {*insultingly, insulting, insulted, insults, insult*}.

The main difficulty lies in finding the division which yields the best value for Q. It is of course infeasible to test each possible division of the network. Newman [19] therefore proposes a method of agglomerative hierarchical clustering starting from communities made of a single node. Communities are repeatedly joined together in pairs, choosing the join that leads to the biggest increase (or slightest decrease) of Q. The best partition of the network in communities corresponds to the biggest value of Q.

Our experiments with the Newman Clustering algorithm have nevertheless shown that it tends to detect bigger communities than wanted, thus decreasing the precision. We have therefore added an additional constraint on possible joins by measuring the density of edges across communities (*cross-community edge density*).

Cross-community edge density between communities A and B is defined as follows:

$$D_{AB} = \frac{\mathsf{number\_of\_edges}(A, B)}{|A| \times |B|} \tag{2}$$

where $\mathsf{number\_of\_edges}(A, B)$ is the number of edges linking nodes in community A to nodes in community B, and $|A|$ and $|B|$ are the number of nodes in community A and B, respectively. The minimum cross-community edge density is fixed by a parameter d whose value ranges from 0 to 1.

5 Morpheme Analyses

After performing clustering, morpheme analyses are obtained based on the word families identified and the transformation rule edges linking words which belong to the same family. First, a representative word is identified for each word family: this is the shortest word in the family; in case of a tie, the most frequent among the shortest words is chosen. The full morpheme analysis for a word form w consists of its family representative and a string representation of the transformation rules that apply to w. The method is detailed in Algorithm 2.

Algorithm 2. Procedure for obtaining the morpheme analyses, given a word family C and the lexical network G

```
$analyses[*] \leftarrow \emptyset$
$subg \leftarrow \mathsf{get\_subgraph}(G, C)$
for edge $(w_1, w_2, rule)$ in $subg$ do
    $analyses[w_1] \leftarrow analyses[w_1] \cup \mathsf{to\_plain\_string}(rule.pattern)$
    $analyses[w_2] \leftarrow analyses[w_2] \cup \mathsf{to\_plain\_string}(rule.repl)$
end for
$rep \leftarrow \mathsf{get\_family\_representative}(C)$
for word $w$ in word family $C$ do
    $analyses[w] \leftarrow analyses[w] \cup rep$
end for
return $analyses$
```

Example 1. Consider for instance the communities obtained for Fig. 1. The representative for the word family {*insulted*;*insulting*;*insult*;*insults*;*insultingly*} is *insult* since it is the shortest word. The complete analyses for the words are the following:

`insultingly`	`insult _ly _ingly`
`insulting`	`insult _ing`
`insulted`	`insult _ed`
`insults`	`insult _s`
`insult`	`insult`

Two transformation rules apply to the word *insultingly*: `^(.+)ly$` → `\1` and `^(.+)ingly$` → `\1`, which are represented in the final analysis as `_ly _ingly`.

6 Evaluation

In this section, we report the results obtained by MorphoNet at Morpho Challenge 2009 competition 1 (linguistic evaluation) and 2 (information retrieval). For all languages, the value of parameter d (cross-community edge density) was empirically set to 0.1 for community detection.

6.1 Morpho Challenge Competition 1

Table 2 contains the results of the linguistic evaluation (competition 1) for MorphoNet and a simple reference method consisting in splitting words into letters. Results are measured in terms of Precision, Recall and F-Measure.

Table 2. Results for competition 1

Language	Method	Precision	Recall	F-Measure	Rank
English	MorphoNet	65.08%	47.82%	55.13%	7 / 14
	letters	3.82%	99.88%	7.35%	
German	MorphoNet	67.41%	30.19%	41.71%	8 / 15
	letters	2.79%	99.92%	5.43%	
Finnish	MorphoNet	63.35%	22.62%	33.34%	9 / 12
	letters	5.17%	99.89%	9.83%	
Turkish	MorphoNet	61.75%	30.90%	41.19%	7 / 14
	letters	8.66%	99.13%	15.93%	
Arabic vowelized	MorphoNet	92.52%	2.91%	5.65%	8 / 12
	letters	50.56%	84.08%	63.15%	
Arabic non vowelized	MorphoNet	90.49%	4.95%	9.39%	6 / 12
	letters	70.48%	53.51%	60.83%	

MorphoNet performs best for English. The lowest results are obtained for Arabic, which is characterized by good precision but very low recall. Most of the participating systems obtained comparably low results for Arabic and none was able to beat the “letters” reference. This could be explained by the following reasons: (i) the datasets provided for Arabic were too small for MorphoNet to perform well and (ii) the analyses required for Arabic were far more complex (in terms of the number of morphemes per word) than for the other languages.

The results also show that MorphoNet consistently obtains better precision than recall, especially in Arabic. The method relies on a list of transformation rules which are automatically acquired in a first step. It is therefore likely that some important rules are missing, leading to low recall. This problem might be solved by performing multiple iterations of rule induction and clustering or by applying rules in a cascaded manner, so that one rule applies to the output of another rule.

Moreover, the procedure for obtaining morpheme analyses is still very coarse, leading to composite morphemes such as `_ingly`. This could easily be improved by detecting and further decomposing such morphemes.

Finally, transformation rules could be weighted by their productivity or their frequency to improve clustering, since some transformation rules might be more reliable than others.

6.2 Morpho Challenge Competition 2

Table 3 summarises the results of the information retrieval (IR) task (competition 2). Results without morpheme analysis (no analysis) are also provided.

Table 3. IR results (mean average precision MAP)

Method	English	German	Finnish
MorphoNet	0.3560	0.3167	0.3668
No analysis	0.3293	0.3509	0.3519

MorphoNet improves the IR results over unanalysed words for English and Finnish, but not for German. While it is difficult to come up with a clear explanation, this might be due to the compounding nature of German. Indeed, the MorphoNet system does not directly cope with compounding for the time being, which might be especially detrimental to the IR task.

7 Conclusions and Future Work

We have described a novel linguistically motivated approach to unsupervised morpheme analysis relying on a network representation of morphological relations between words. Due to the underlying network representation, it is possible to use community detection and ranking methods devised for other kinds of data. This approach is still in its very early stage, yet the results obtained at Morpho Challenge 2009 demonstrate that it yields very promising results and thus deserves further investigation.

The method described in this paper can be considered as a baseline for network-based morphology induction. It leaves lots of room for improvement. A first objective would be to obtain a better recall for morpheme analysis. This necessitates to provide a better mechanism for the acquisition of transformation rules. It should be possible to perform multiple iterations of the rule induction and clustering cycle or to apply rules in a cascaded manner. This is especially needed for languages which are morphologically more complex than English such as Turkish or Finnish. Also, we have not weighted the edges in the graph, which could be useful to improve clustering.

The clustering method performs hard-clustering: each word belongs to only one family. This is especially detrimental for languages like German, for which it would be desirable to allow multiple family membership in order to take compounding into account. In the future, we would therefore like to better address compounding.

Graphs also open up the way for a new form of modelisation of morphology enabling the analysis of crucial morphological properties. In particular, node properties in the graph could be used to rank nodes and better detect base words in families, using algorithms such as PageRank.

Acknowledgements

We are grateful to the anonymous reviewers for their helpful comments.

References

1. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview and Results of Morpho Challenge 2009. In: Multilingual Information Access Evaluation 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Revised Selected Papers. LNCS, vol. I. Springer, Heidelberg (2010)
2. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph-based Word Clustering using a Web Search Engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 542–550 (2006)
3. Mihalcea, R.: Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algo-rithms for Sequence Data Labeling. In: Proceedings of the HLT/EMNLP 2005 Conference, pp. 411–418 (2005)
4. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of EMNLP 2004, pp. 404–411 (2004)
5. Bybee, J.: Morphology: A Study of the Relation between Meaning and Form. Benjamins, Philadelphia (1985)
6. Neuvel, S., Fulop, S.A.: Unsupervised Learning of Morphology Without Morphemes. In: Proceedings of the ACL Workshop on Morphological and Phonological Learning 2002, pp. 31–40 (2002)
7. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69 (2004)
8. Creutz, M., Lagus, K.: Unsupervised Discovery of Morphemes. In: Proceedings of the ACL Workshop on Morphological and Phonological Learning 2002, pp. 21–30 (2002)
9. Creutz, M., Lagus, K.: 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 2005 (2005)
10. Bernhard, D.: Apprentissage non supervisé de familles morphologiques par classification ascendante hiérarchique. In: Actes de la 14e conférence sur le Traitement Automatique des Langues Naturelles TALN 2007, pp. 367–376 (2007)
11. Hathout, N.: Acquistion of the Morphological Structure of the Lexicon Based on Lexical Similarity and Formal Analogy. In: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing (COLING 2008), pp. 1–8 (2008)
12. Bernhard, D.: Unsupervised Morphological Segmentation Based on Segment Predictability and Word Segments Alignment. In: Proceedings of the Pascal Challenges Workshop on the Unsupervised Seg-mentation of Words into Morphemes, pp. 19–23 (April 2006)
13. Dasgupta, S., Ng, V.: High-Performance, Language-Independent Morphological Segmentation. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007), pp. 155–163 (2007)
14. Demberg, V.: A Language-Independent Unsupervised Model for Morphological Segmentation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 920–927 (2007)
15. Baroni, M., Matiasek, J., Trost, H.: Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In: Proceedings of the ACL Workshop on Morphological and Phonological Learning 2002, pp. 48–57 (2002)

16. Gaussier, E.: Unsupervised learning of derivational morphology from inflectional lexicons. In: Proceedings of the Workshop on Unsupervised Methods in Natural Language Processing (1999)
17. Schone, P., Jurafsky, D.: Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. In: Proceedings of the Fourth Conference on Computational Natural Language Learning, pp. 67–72 (2000)
18. Hathout, N.: From WordNet to CELEX: acquiring morphological links from dictionaries of synonyms. In: Proceedings of the Third International Conference on Language Resources and Evaluation, pp. 1478–1484 (2002)
19. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E 69 (2004)
20. van Dongen, S.: Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht (2000)
21. Dorow, B., Widdows, D., Ling, K., Eckmann, J.P., Sergi, D., Moses, E.: Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. In: 2nd MEANING Workshop (2005)