Simple Morpheme Labelling in Unsupervised Morpheme Analysis

Delphine Bernhard

Ubiquitous Knowledge Processing Lab
Computer Science Department
Technische Universität Darmstadt, Germany
delphine@tk.informatik.tu-darmstadt.de

Abstract. This paper describes a system for unsupervised morpheme analysis and the results it obtained at Morpho Challenge 2007. The system takes a plain list of words as input and returns a list of labelled morphemic segments for each word. Morphemic segments are obtained by an unsupervised learning process which can directly be applied to different natural languages. Results obtained at competition 1 (evaluation of the morpheme analyses) are better in English, Finnish and German than in Turkish. For information retrieval (competition 2), the best results are obtained when indexing is performed using Okapi (BM25) weighting for all morphemes minus those belonging to an automatic stop list made of the most common morphemes.

1 Introduction

The goal of Morpho Challenge 2007 [1,2] was to develop algorithms able to perform unsupervised morpheme analysis, which consists in automatically discovering a word's morphemes using only minimal resources made up of a list of words and a text corpus in each language. Morphemic segments have to be identified but also labelled and hence disambiguated. On the one hand, morphemes are abstract units which may be realised by several surface forms, i.e. allomorphs. On the other hand, a single surface form may correspond to several homographic morphemes. In order to perform morpheme analysis and correctly label morphemic segments, it is therefore necessary to achieve a mapping between morpheme labels and their surface realisations.

As it will be evidenced later, the system presented in this paper does not solve cases of allomorphy since different surface forms will always be considered as different morphemes. It only partly aims at resolving cases of homography since identified morphemic segments are labelled with one of the following categories: stem/base, prefix, suffix and linking element. The system nevertheless achieved decent results in competition 1 for English, Finnish and German, but results in Turkish were less satisfactory.

The rest of this paper is organised as follows. The algorithm is described in Section 2. Results obtained for competitions 1 and 2 are presented in Section 3. Then, particular assumptions made in the system and pertaining to morpheme

C. Peters et al. (Eds.): CLEF 2007, LNCS 5152, pp. 873–880, 2008.

labelling are discussed in Section 4. Finally perspectives for the evolution of the system are given in Section 5.

2 Overview of the Method

The algorithm is mostly identical to the one already presented at Morpho Challenge 2005, apart from a few minor changes. A previous and more detailed description of the system can be found in [3]. The method takes as input a plain list of words without additional information. The output is a labelled segmentation of the input words. Labels belong to one of the following categories: stem, prefix, suffix and linking element. The algorithm can be subdivided into 4 main steps, plus an additional step which may be performed to analyse another data set, given the list of segments learned after step 4.

2.1 Step 1: Extraction of Prefixes and Suffixes

The objective of step 1 is to acquire a list of prefixes and suffixes. The longest words in the input word list are segmented based on the notion of segment predictability. This idea is recurrent in research on the segmentation of words into morphemes (see for instance [4,5,6]). It posits that a morpheme boundary can be hypothesised if it is difficult to predict the character (or string of characters) which follows, knowing an initial string of characters. In the system, segment predictability is modelled by computing the average maximum transition probabilities between all the substrings of a word coalescing at a given position k within the word. The variations of this measure make it possible to identify morpheme boundaries at positions where the average maximum transition probabilities reach a well-marked minimum. Figure 1 depicts the variations of the average maximum transition probabilities for the English word "hyperventilating". Two morpheme boundaries are identified in this word, which corresponds to the following segmentation: "hyper + ventilat + ing".

Once a word has been segmented in this fashion, the longest and less frequent amongst the proposed segments is identified as a stem, if this segment also appears at least twice in the word list and at least once at the beginning of a word. In the example of Fig. 1, the segment 'ventilat' will be identified as a valid stem.

The identified stem is then used to acquire affixes. All the substrings preceding this stem in the input word list are added to the list of prefixes unless they are longer and less frequent than the stem. Correspondingly, all the substrings following this stem in the word list are added to the list of suffixes unless they are longer and less frequent than the stem. Moreover, one character-long prefixes are eliminated because these often lead to erroneous segmentations in later stages of the algorithm.

This procedure is applied to the longest words in the input word lists. The process of affix acquisition ends when for N running words the number of new affixes among the affixes learned is inferior to the number of affixes which already belong to the list of prefixes and suffixes.

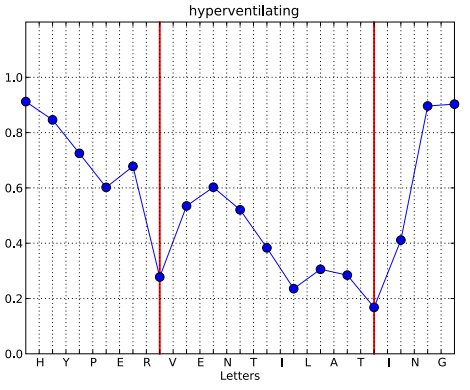

Fig. 1. Variations of the average maximum transition probabilities between substrings of the word "hyperventilating". Boundaries are marked with a bold vertical line.

2.2 Step 2: Acquisition of Stems

The aim of the second step is to acquire a list of stems, using the prefixes and suffixes which have been previously identified. Stems are obtained by stripping off from each word in the input word list all the possible combinations of prefixes, suffixes and the empty string. In order to reduce the noise induced by such a simple method, some constraints are applied, especially a minimum length threshold of 3 characters for the stem. Note that the stems acquired by this method are not all minimal and may still contain affixes.

2.3 Step 3: Segmentation of Words

In a third step, all the words are segmented. Word segmentation is performed by comparing words which contain the same stem to one another. This consists in finding limits between shared and different segments and results in a segmentation of the words being compared. The outcome can be represented as an alignment graph for each stem. Figure 2 depicts the alignment graph obtained for the words containing the English stem 'integrat'.

Segments are subsequently labelled with one of the three non-stem types (prefix, suffix, linking element) according to their positions within the word, relatively to the stem. As a result of the alignment, prefixes and suffixes which do not belong to the list of affixes acquired after step 1 may be discovered. A validation procedure, similar to the one proposed by [4], is therefore applied. It consists in checking that the proportion of new affixes in the alignment graph does not exceed some threshold[1]. All the segmentations made up of valid morphemic segments are stored.

[1] There are actually two different thresholds, a and b. For details about these thresholds, see [3].

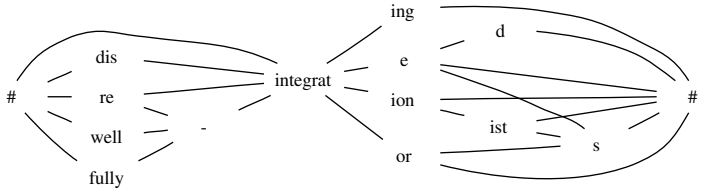

Fig. 2. Example segmentation of words sharing the stem "integrat"

2.4 Step 4: Selection of the Best Segmentation

As a result of Step 3, several different segmentations may have been discovered for each word, since a word may contain more than one potential stem. In order to select the best possible segments, a best-first search strategy is applied on the potential segments of a word, privileging the most frequent segment when given a choice. Some frequency and morphotactic constraints are also checked (e.g. a prefix cannot be directly followed by a suffix, there has to be at least one stem in the proposed segmentation, etc.).

2.5 Step 5: Application of the Learned Segments to a New Data Set (Optional)

The morphemic segments identified after Step 4 can be used to segment any list of words or the whole list of words when segments are learned using only a subset of the list. An A*-like algorithm is used to find the best segmentation for each word, i.e. the segmentation with the lowest global cost which also complies to morphotactic constraints similar to those used at step 4. The global cost for a segmentation is the sum of the costs associated with each segment s_i. Two different segment cost functions have been used resulting in two different submissions for each language:

$$\text{cost}_1(s_i) = -\log \frac{f(s_i)}{\sum_i f(s_i)} \tag{1}$$

$$\text{cost}_2(s_i) = -\log \frac{f(s_i)}{\max_i[f(s_i)]} \tag{2}$$

where $f(s_i)$ is the frequency of s_i.

3 Morpho Challenge 2007 Experiments and Results

The method has been applied to all of the four test languages of Morpho Challenge 2007. Morphemic segments have been learned using only a subset of the word lists provided for competition 1 (the 300,000 most frequent words), without taking into account contextual information found in the text corpora. These

morphemic segments have then been used to segment all the words in the data sets provided both for competition 1 and 2, using either cost_1 or cost_2.

Moreover, no fine tuning of the three different parameters of the system (N, a and b) has been attempted. Earlier experiments have shown that default values N=5, a=0.8 and b=0.1 are globally reasonable and these have consequently been used for all four languages.

3.1 Results for Competition 1: Morpheme Analysis

In competition 1, the system's analyses have been compared to a linguistic gold standard in Finnish, Turkish, German and English [1]. Table 1 details the precision, recall and F-measure obtained by the system. Method 1 corresponds to results obtained using cost_1 and method 2 to results obtained using cost_2.

Table 1. Precision %, recall % and F-measure % obtained for competition 1

Language	Method 1			Method 2		
	Precision	Recall	F-measure	Precision	Recall	F-measure
English	72.05	52.47	60.72	61.63	60.01	**60.81**
Finnish	75.99	25.01	37.63	59.65	40.44	**48.20**
German	63.20	37.69	47.22	49.08	57.35	**52.89**
Turkish	78.22	10.93	19.18	73.69	14.80	**24.65**

As it had already pointed out at Morpho Challenge 2005, results for method 1, using cost_1, indicate higher precision but lower recall on all datasets. On the whole, better F-measures are obtained with method 2 for all languages. Results in Turkish are well under those obtained in the other languages and are characterised by very poor recall. The recall of most of the other systems which also took part in Morpho Challenge 2007 is below 20% as well for this particular language. One possible explanation for this is that there are more different analyses per word in Turkish (at least in the provided gold standard sample), and therefore more ambiguities have to be solved in the proposed morpheme analyses than in the other languages.

3.2 Results for Competition 2: Information Retrieval

In competition 2, the morphological analyses were used to perform information retrieval experiments. The experimental set-up is described in detail in [2]. Table 2 lists the results obtained for the information retrieval task.

The results obtained by the algorithm strongly depend on the weighting scheme used and are better with the Okapi BM25 weighting and a stop list, whatever the method and the word list used. Moreover, while method 1 performs slightly better than method 2 with the tf-idf weighting, this tendency is reversed with the Okapi weighting (except for German). It is not clear how this could be accounted for but a possible explanation is that method 2, which is less precise, benefits more than method 1 from the removal of the most frequent morphemes.

Table 2. Average precision obtained for competition 2

	English		Finnish		German	
	tf-idf	Okapi	tf-idf	Okapi	tf-idf	Okapi
method 1 - without new	0.2781	0.3881	0.4016	0.4183	0.3777	0.4611
method 1 - with new	0.2777	0.3900	0.3896	0.4681	0.3720	**0.4729**
method 2 - without new	0.2673	0.3922	0.3984	0.4425	0.3731	0.4676
method 2 - with new	0.2682	**0.3943**	0.3811	**0.4915**	0.3703	0.4625

4 Analysis and Discussion

As mentioned in the introduction, the main objective of Morpho Challenge 2007 is to obtain a morpheme analysis of the word forms, which is a lot more demanding than just segmenting words into morphs. A morpheme analysis for a word corresponds to a list of labelled morphemes. A minimal morpheme analysis may consist of a list of unlabelled morphemic segments identified after morphological segmentation. The algorithm presented in this paper corresponds to an intermediary and simple solution since it labels the segments with general morpheme categories, which are detailed in the next section.

4.1 Morpheme Categories

The morphemic segments discovered by the system are labelled by one of the following categories: stem or base (B), prefix (P), suffix (S) and linking element (L). Table 4.1 gives some examples of the labelled morpheme analyses produced by the system compared with the gold standard analyses.

The basic base, prefix and suffix categories are taken into account by several systems which perform unsupervised morphological analysis such as the Morfessor Categories systems [7,8]. The *linking element* category is intended to encompass short segments (usually just one letter long) which link two words or word-forming elements in compounds, such as hyphens, neo-classical linking elements or German *Fugenelemente*. Linking elements differ from the other categories of morphemes because they bring no semantic contribution to the overall meaning of the word.

Table 3. Example morpheme analyses

	Word	Method 1	Method 2	Gold standard
Eng.	chilly	chill_B y_S	chill_B y_S	chill_A y_s
	planners'	planner_B s_S '_S	plann_B er_S s'_S	plan_N er_s +PL +GEN
Fin.	ikuisuus	ikuis_B uus_B	ikuis_B uu_L s_S	ikuinen_A +DA-UUS
	resoluutio	resoluutio_B	resoluutio_B	resoluutio_N
Ger.	bezwingen	be_P zwing_B en_S	be_P zwing_B e_S n_S	be zwing_V +13PL
	risikoloser	risiko_B los_S er_S	risiko_B los_S er_S	risiko_N los +ADJ-er
Tur.	avucuna	a_P v_P ucu_B na_S	a_P v_P ucu_B na_S	avuc +POS2S +DAT
	kolaCan	kol_B aCan_B	kol_B aCan_B	kolaCan

4.2 Accuracy of Morpheme Labelling

As stated in the introduction, a further objective of morpheme labelling is to disambiguate cases of allomorphy and homography. The recognition of allomorphy is beyond reach of the system in its current state. For instance, the allomorphs of the English prefix in+ (im-, in-, ir-) or of the suffix +able (-able, -ible) will always be recognised as different morphemes.

Homography should be partially dealt with by the system when homographs belong to different morphemic categories, which excludes within-category homography as squash_N and squash_V where "squash" will only be identified as a stem.

In order to verify this assumption, let us consider the example of the segment 'ship' in English. This segment is either a stem (meaning 'vessel') or a suffix which refers to a state. The segment 'ship' is correctly labelled as a suffix by method 1 in words like "censorship" (censor_B ship_S) or "citizenship" (citizen_B ship_S). The stem 'ship' can be correctly identified either by method 1 or 2 when it is found at the beginning of a word, but not when it is found at the end of a word; for instance, "shipwreck" is analysed as ship_B wreck_B by methods 1 and 2, while "cargo-ship" is analysed as cargo_B -_L ship_S by method 1.

The previous examples reveal that the simple morpheme labelling performed by the system does not solve all detectable cases of homography between stems and affixes. Morphotactic constraints help in that respect, since they prevent a suffix from occurring at the beginning of a word, and thus the suffix 'ship' will not be identified at word initial positions. However, the final analysis privileges the most frequent segment, when several morpheme categories are morphotactically plausible. This tends to be favourable to affixes since they are usually more frequent than stems.

5 Future Work

As shown in the previous section, morphotactic constraints, as they are currently used in the system, are not always sufficient and flexible enough to disambiguate between several homographic segments. For the time being, these constraints are implemented as a simple deterministic automaton, which is the same for all languages. In the future versions of the system, it would be desirable to bootstrap these constraints from the data themselves, as suggested by [9].

Another direction for future research concerns the integration of corpus-derived information. Several algorithms have demonstrated the usefulness of contextual information for unsupervised morphological analysis, to complement orthographic information with semantic and syntactic constraints. Corpus-derived knowledge can be used either at the beginning of the process [10,11], or at the end [12]. In the first case, only words which are contextually similar are compared to discover morphemes. In the second case, spurious morphological analyses are filtered out by taking semantic similarity into account. Corpus-derived information could be incorporated in the first step of the current algorithm, in order to increase the precision of affix acquisition since most of the subsequent processes rely on the affixes

acquired at step 1. Also, it is obviously worth investigating the use of text corpora to achieve finer-grained morpheme labelling and refine the very general categories used so far.

References

1. Kurimo, M., Creutz, M., Varjokallio, M.: Morpho Challenge Evaluation using a Linguistic Gold Standard. In: Peters, C., et al. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 864–873. Springer, Heidelberg (2008)
2. Kurimo, M., Creutz, M., Turunen, V.: Morpho Challenge Evaluation by IR Experiments. In: Proceedings of the CLEF 2007 Workshop. LNCS. Springer, Heidelberg (2008)
3. Bernhard, D.: Unsupervised Morphological Segmentation Based on Segment Predictability and Word Segments Alignment. In: Kurimo, M., Creutz, M., Lagus, K. (eds.) Proceedings of the Pascal Challenges Workshop on the Unsupervised Segmentation of Words into Morphemes, Venice, Italy, pp. 19–23 (April 2006)
4. Déjean, H.: Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. In: Powers, D. (ed.) Proceedings of the CoNLL98 Workshop on Paradigms and Grounding in Language Learning, pp. 295–298 (1998)
5. Hafer, M.A., Weiss, S.F.: Word segmentation by letter successor varieties. Information Storage and Retrieval 10, 371–385 (1974)
6. Harris, Z.: From phoneme to morpheme. Language 31(2), 190–222 (1955)
7. Creutz, M., Lagus, K.: Induction of a Simple Morphology for Highly-Inflecting Languages. In: Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON), Barcelona, pp. 43–51 (2004)
8. 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), Espoo, Finland, pp. 106–113 (2005)
9. Demberg, V.: A Language-Independent Unsupervised Model for Morphological Segmentation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp. 920–927. Association for Computational Linguistics (2007)
10. Bordag, S.: Two-step Approach to Unsupervised Morpheme Segmentation. In: Proceedings of the Pascal Challenges Workshop on the Unsupervised Segmentation of Words into Morphemes, Venice, Italy, pp. 25–29 (2006)
11. Freitag, D.: Morphology Induction from Term Clusters. In: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL 2005), Ann Arbor, Michigan, pp. 128–135. Association for Computational Linguistics (2005)
12. Schone, P., Jurafsky, D.: Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. In: Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, Portugal (September 2000)