

Unsupervised Learning of Morphology by Using Syntactic Categories

Burcu Can, Suresh Manandhar
Department of Computer Science, University of York
York YO10 5DD UK
{burcu,suresh}@cs.york.ac.uk

Abstract

This paper presents a method for unsupervised learning of morphology that exploits the syntactic categories of words. Previous research [4][12] on learning of morphology and syntax has shown that both kinds of knowledge affect each other making it possible to use one type of knowledge to help the other. In this work, we make use of syntactic information i.e. Part-of-Speech (PoS) tags of words to aid morphological analysis. We employ an existing unsupervised PoS tagging algorithm for inducing the PoS categories. A distributional clustering algorithm is developed for inducing morphological paradigms.

Categories and Subject Descriptors

1.2 [Artificial Intelligence]: 1.2.6 Learning 1.2.7 Natural Language Processing

General Terms

Algorithms, Experimentation

Keywords

Morphological Analysis, Syntax, Unsupervised Learning, Clustering

1 Introduction

Morphological analysis is a crucial subtask for most Natural Language Processing (NLP) problems: Machine Translation, Question Answering, Information Retrieval, Part-of-Speech Tagging etc. Morphological segmentation is a preprocessing step before a full morphological analysis can be done. For applications dealing with large corpora morphological segmentation requires dealing with sparseness of data. For example, a verb can have thousands of different forms in languages like Finnish (Karlsson [13]) and often all the different forms are not seen during training.

Morphology learning has been an active research topic since Harris [11] proposed the distributional hypothesis. One of the earliest works on morphology learning using Harris's ideas is Déjean and Norm's method [8]. In this work [8], Déjean and Norm find the most frequent morphemes on the given unannotated text, and by using this morpheme list, he segments the given text morphologically. Since then, there has been huge amount of progress made in the field. We can classify these into the following approaches: Minimum Description Length (MDL) model, Letter Successor Variety (LSV) model, Semantic models (ex: Latent Semantic Analysis - LSA), Probabilistic

models.

MDL based models (Brent [2], Brent et. al. [3], Goldsmith [9][10], Creutz and Lagus [7]) aim to minimize the space used by the corpus and the model by morphologically segmenting the words in the corpus. LSV model has been employed by Bordag [1] that used letter frequencies to find split points in the words. Snover et. al. [16] describe a probabilistic model in which morphological paradigms are created gradually by choosing the number of stems, morphemes, paradigms in a probabilistic, and generative manner. Another generative model is due to Creutz [6] in which the lengths and the frequencies of the morphemes are used as prior information. Schone and Jurafsky [15] used LSA to capture the semantic relatedness of words to aid morphological segmentation.

All the above work has been primarily employed to learn simple i.e. non-recursive concatenative morphology but they do not directly address the recursive nature of the morphology of agglutinative languages. Monson [14] proposed a system for handling the morphology of agglutinative languages. His system achieved a precision of 52% on Turkish as evaluated in the Morpho Challenge Workshop, 2008.

There has been some work on the joint unsupervised learning of morphology and PoS tags. Hu et. al. [12] extends the minimum description length (MDL) based framework due to Goldsmith [9] to explore the link between morphological signatures and PoS tags. Clark and Tim [4] experiments with the fixed endings of the words for PoS clustering. So morphologically similar words tend to belong into the same PoS cluster.

Our current work can be viewed is in a similar direction. In particular, we show that unsupervised PoS tagging can be effectively employed for learning of morphology. However, the work presented here is not a method for simultaneous learning of PoS categories and morphology. It is limited to learning of morphology given that PoS categories already been induced using an unsupervised PoS tagger.

2 Inducing Syntactic Categories

For the induction of syntactic categories, we used Clark’s distributional clustering approach [5] which can be considered as an instance of *average link clustering*. Although it should be emphasised that any other method for unsupervised induction of PoS tags can be substituted without affecting the method presented in this paper. Following Clark’s approach, each word is clustered by using its context. A *context* consists of the previous word, and the following word. Each word has a context distribution over all ordered pairs of left-context/right-context words. To measure the distributional similarity between words, KL divergence is used which is defined as:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (1)$$

where p, q are the context distributions of the words being compared and x ranges over contexts.

In his approach [5], the probability of a context for a target word is defined as:

$$p(\langle w_1, w_2 \rangle) = p(\langle c(w_1), c(w_2) \rangle) p(w_1|c(w_1)) p(w_2|c(w_2)) \quad (2)$$

where $c(w_1), c(w_2)$ denote the PoS cluster of words w_1, w_2 respectively.

The algorithm requires the number of clusters K to be specified in advance. In addition to the K clusters, one spare cluster is employed containing all unclustered words. During each iteration, one word is chosen from the spare cluster having the minimum KL divergence with one of the K

clusters. For each cluster, its context distribution is computed as the averaged distribution of all words in the cluster. In addition, the KL divergence between clusters are computed after each iteration and clusters are merged if the divergence is below a manually set threshold.

We set $K=77$, the number of tags defined in CLAWS tagset used for tagging the BNC (British National Corpus). We used the same number of clusters for Turkish and German. Final clusters show that PoS clusters are related with the major syntactic categories. The system finds PoS clusters that can be identified as proper nouns, verbs in past tense form, verbs in present continuous form, nouns, adjectives, adverbs, and so on.

Some sample clusters are given below for English:

Cluster 1: much far badly deeply strongly thoroughly busy rapidly slightly heavily neatly widely closely easily profoundly readily eagerly

Cluster 2: made found held kept bought heard played left passed finished lost changed etc

Cluster 3: should may could would will might did does etc

Cluster 4: working travelling flying fighting running moving playing turning etc

Cluster 5: people men women children girls horses students pupils staff families etc

3 Inducing Morphological Paradigms

After the induction of the syntactic categories, the conditional probability of each morpheme x given its PoS cluster c , $p(x|c)$, is computed. The possible morphemes in each PoS cluster is found by splitting each word in each cluster at all split points, and the morphemes are ranked, and sorted according to their maximum likelihood estimates in each PoS cluster.

A list of highest ranked morphemes are given in Table 1 for English, German, and Turkish.

Table 1: Some high ranked potential morphemes in PoS clusters

English		German		Turkish	
Cluster	Morphemes	Cluster	Morphemes	Cluster	Morphemes
1	-s	1	-n,-en	1	-i,-si,-ri
2	-d,-ed	2	-e,-te	2	-mak,-mek,-mesi,-masi
3	-ng,-ing	3	-g,-ng,-ung	3	-an,-en
4	-y,-ly	4	-r,-er	4	-r,ar,er,-ler,-lar
5	-s,-rs,-ers	5	-n,-en,-rn,-ern	5	-r,-ir,-dir,-Ir,-dIr
6	-ing,-ng,g	6	-ch,-ich,-lich	6	-e,-a

This ranking is used to eliminate the potential non-morphemes with a low conditional probability hence reducing the search space. In the next step, morphemes across PoS clusters are incrementally merged forming the basis of the paradigm capturing mechanism. In each iteration, a morpheme pair across two different PoS cluster with the highest number of common stems is chosen for merging. Once a morpheme pair is merged the words that belong to this newly formed paradigm are removed from their respective PoS clusters. Once a word is assigned to a paradigm, it cannot be part of any other paradigm. Thus, we postulate that a word can only belong to a

single morphological paradigm.

Since, in our current framework, morphemes are tied to PoS clusters our definition of paradigm deviates from that of Goldsmith [9] in that a paradigm ϕ is a list of morpheme/cluster pairs i.e. $\phi = \{m_1/c_1, \dots, m_n/c_n\}$. Associated with each paradigm is a list of stems i.e. the list of stems that can combine with each of the morphemes m_i to produce a word belonging to the c_i PoS category.

Algorithm 1 describes the complete paradigm capturing process. Some examples of sample paradigms captured are given below:

English:

ed ing : reclaim aggravat hogg trimm expell administer divert register stimulat shap rehabilitat exempt stiffen spar deceiv contaminat disciplin implement stabiliz feign mistreat extricat mimick alert seal etc

s d : implicate ditche amuse overcharge equate despise torpedoe curse plie supersede preclude snare tangle eclipse relinquishe ambushe reimburse alienate conceive vetoe waive envie negotiate diagnose etc

er ing : brows wring worship cropp cater stroll zipp moneymak tun chok hustl angl windsurf swindl cricket painkill climb heckl improvis scream scaveng panhandl lawmak bark clean lifesav beekeep toast matchmak bodybuild etc

e ed : subsid liquidat redecorat exorcis amputat fertiliz reshap regulat foreclos infring eradicate reverberat chim centralis restructur crippl rehabilitat symbolis reinstat etc

ly er : dark cheap slow quiet fair light high poor rich cool quick broad deep bright calm crisp mild clever etc

0 s : benchmark instrument pretzel wheelchair scapegoat spike infomercial catastrophe beard paycheck reserve abduction

Turkish:

i e : zemin faaliyetin torenler secim incelemeler eyalet nem takvim makineler yontemin becerisin gorusmeler teknigin merkezin iklim goruntuler etc

i a : cevab bakimin mektuplar esnaf olayin akisin miktar kayd yasamay bulgular sular masraflarin heyecanin kalan haklarin anlamin etc

i in : sanayiın degerlerin esin denizler duman teminat erkekler kurullarin birbirin vatandaslarimiz gelismesin milletvekillerin partisin

de e : bolgesin duzeyin yonetimin dergisin sektorun birimlerin bolgelerin tumun bolumlerin tesislerin donemin kongresin evin etc

mesi en : izlen yurutul degis uretil gerceklestiril desteklen gelistiril etc

i 0 : iman cekim mahkemelerin orneklem gaffet yazman sanat trendler mahalleler eviniz hamamlar piller ogretim olimpiyat

German:

r n : kurze ehemalige eidgenoessische professionelle erste bescheidene ungewoehnliche ethnische unbekante besondere nationalsozialistische deutsche

e en : praechtig gesichert dauerhaft bescheiden vereinbart biologisch natuerlich oekumenisch kantonal unterirdisch wissenschaftlich nahegelegen chinesisich

t en : funktionier konkurrier schneid mitwirk ansteig plaedier pfeif aufklaer schluck ausgleich weitermach abhol ankomm spazier speis aussteig aufhoer

er ung : versteiger unterdrueck erneuer vermarkt beschleunig besetz geschaeftsfuehr wirtschaftsfoerder finanzverwalt verhandl

s 0 : potential instrument flohmarkt vorhang pilotprojekt idol rechner thriller ensemble bebauungsplan empfinden defekt aufschwung

Algorithm 1 Algorithm for paradigm capturing using syntactic categories

```
1: Apply unsupervised PoS clustering to the input corpus
2: For each PoS cluster  $c$  and morpheme  $m$ , compute maximum likelihood estimates of  $p(m | c)$ 
3: Keep all  $m$  (in  $c$ ) with  $p(m | c) > t$ , where  $t$  is a threshold
4: repeat
5:     for all PoS clusters  $c_1, c_2$  do
6:         Pick morphemes  $m_1$  in  $c_1$  and  $m_2$  in  $c_2$  with the highest number of
           common stems
7:         Store  $\phi = \{m_1/c_1, m_2/c_2\}$  as the new paradigm
8:         Remove all words in  $c_1$  with morpheme  $m_1$  and associate these words
           with  $\phi$ .
9:         Remove all words in  $c_2$  with morpheme  $m_2$  and associate these words
           with  $\phi$ .
10:    end for
11:    for all Paradigms  $\phi_1, \phi_2$  such that  $Acc(\phi_1, \phi_2) > T$ , where  $T$  is a threshold do
12:        Create new merged paradigm  $\phi = \phi_1 \cup \phi_2$ 
13:        Associate all words from  $\phi_1$  and  $\phi_2$  into  $\phi$ 
14:        Delete paradigms  $\phi_1, \phi_2$ .
15:    end for
16: until No morpheme pair consisting of at least one common stem is left
```

4 Merging Paradigms

For capturing more general paradigms, paradigm merging is performed. We rank potential paradigms by the ratio of common stems with the total number of stems captured by the paradigm. More precisely, given paradigms ϕ_1, ϕ_2 , let P be the total number of common stems. Let N_1 be the total number of stems in ϕ_1 that are not present in ϕ_2 . Similarly, let N_2 be the total number of stems in ϕ_2 that are not present in ϕ_1 . Then, we can define the expected paradigm accuracy of ϕ_1 with respect to ϕ_2 by:

$$Acc_1 = \frac{P}{P + N_1} \quad (3)$$

Acc_2 is defined analogously.

We use the average of Acc_1 and Acc_2 to compute the combined (averaged) expected accuracy of the merged paradigms ϕ_1, ϕ_2 :

$$Acc(\phi_1, \phi_2) = \frac{\frac{P}{P+N_1} + \frac{P}{P+N_2}}{2} \quad (4)$$

During each iteration, all the paradigm pairs having an expected accuracy greater than a given threshold value are merged. Once two paradigms are merged, stems that occur in only one of the paradigms inherit the morphemes from the other paradigm. This mechanism helps create a more general paradigm and helps recover missing word forms. Thus, although some of the word forms do not exist in the corpus, it becomes possible to capture these forms.

Some example paradigms that are found by the system are given below:

English:

es ing e ed: sketch chew nipp debut met factor profit occur err trudg participat necessitat stomp streak siphon stroll sprint drizzl firm climax gestur whipp roll tripp stemm dangl shuffl kindl broker chalk latch rippl collaborat chok summ propp pedal paralyz parad plough cramm slack wad saddl conjur tipp gallop totall catalogu bundl barg whittl retaliat straighten tick peek jabb slimm

s ing ed 0: benchmark mothball weed snicker thread queue jack paw yacht implement import bracket whoop conflict spoof stunt bargain honor bird fingerprint excerpt handcuff veil comment

Turkish:

u a e i : yapabileceklerin kredisin hizmetleri'n sevdikleriniz yeter' transferlerin sevin elimiz tehlikelerin sas mucizey tehditlerin bakir muhasebesin ed gayrimenkuller ecevit' defterim izlemelerin tescilin minarey tahsilin lastikler yerlestirmey

i lar li in : ruhsat semt ikilem reaksiyonlar harc tip prim gidilmis kaldirmis degistirmis bulunmayacak aktarmis bulunacak kapanacak yazilabilecek devredilmis degisecek gelmemis

German:

er 0 e en: kassiert beguenstigt eingeholt genuegt angelastet beruehrt beinhaltet zurueckgegeben beschleunigt initiiert abgestellt bewirkt mitgenommen abgebrochen beruhigt besichtigt

te ung er ten t en lich e : fahr gebrauch blockier identifizier studier entfalt gestalt agier passier sprech berat tausch kauf such weck beug erreich bearbeit beobacht erleid ueberrasch halt helf oeffn pruef uebertreff bezahl spring fuell toet

0 te t er : lichtenberg limburg hill trier elmshorn dreieich praunheim heusenstamm heddernheim hellersdorf schmitt muehlheim lueneburg kassel schluechtern preungesheim rodgau bieber osnabrueck rodheim muenchen london lissabon seoul wedding treptow

5 Morphological Segmentation

For Morpho Challenge 2009, we first clustered all the words in the given corpora thereby creating a set of PoS clusters. We then followed the steps described in the previous sections to induce the morphological paradigms.

Wordlists as provided in Morpho Challenge 2009 contain the list of words that need to be segmented. To assign a PoS cluster to given word, w from the wordlist, the context distribution of w is first computed. The word is assigned the PoS cluster with the minimal KL divergence. In this case, we only consider words with a frequency greater than 10 to eliminate noise.

To segment the words in the word lists, first the word is checked if it exists in one of the existing paradigms. We followed different algorithms for known, unknown and compound words:

5.1 Handling known words

If the word exists in one of the paradigms, it is segmented by using the morpheme in the paradigm in which the word is found. For example, if a paradigm exists as given below:

s ing ed 0 : benchmark mothball weed snicker thread queue jack paw yacht implement import bracket whoop

If a word 'importing' is to be morphologically analyzed, it is automatically segmented by using the morpheme 'ing'.

5.2 Handling unknown words

If the word does not exist in any of the paradigms, a sequence of segmentation rules are applied. By using the paradigms, we created a morpheme dictionary to split the words which do not belong to any of the paradigms. All the morphemes in each paradigm are included in the morpheme dictionary if in any of the paradigm the initial letters of the morphemes are not the same. If the

initial letters of the all morphemes in the same paradigm are the same, the longest morpheme is included in the dictionary. Using the morpheme dictionary, the word is scanned from the rightmost letter to check if any of the endings of the word exist in the dictionary. The longest letter sequence (of the word) existing in the dictionary is chosen to split the word. The same process is repeated after splitting the word until no split can be applied.

5.3 Handling compounds

For the compounds, such as 'hausaufhaben' in German, or 'railway' in English, for both known, and unknown words a recursive approach is performed. The compounding rules split a word recursively from the rightmost end to the left. If an ending sequence of letters exists as a word in the corpus, the sequence is split, and the same procedure is repeated until no valid internal word part is a valid word itself in the corpus. When there are multiple matches the longest match is chosen. This recursive search is also able to find the prefixes as it searches for the valid sub-words in the words.

Algorithm for the segmentation of the words is given in Algorithm 2.

Algorithm 2 Morphological Segmentation

```

1: for all For each given word,  $w$ , to be segmented do
2:   if  $w$  already exists in a paradigm  $\phi$  then
3:     Split  $w$  using  $\phi$  as  $w = u + m$ 
4:   else
5:      $u = w$ 
6:   end if
7:   If possible split  $u$  recursively from the rightmost end by using the morpheme dictionary
   as  $u = s_1 + \dots + s_n$  otherwise  $s_1 = u$ 
8:   If possible split  $s_1$  into its sub-words recursively from the rightmost end as  $s_1 = w_1 +$ 
    $\dots + w_n$ 
9: end for

```

6 Results

The model is evaluated in Morpho Challenge 2009 competition. Here we describe the datasets we used, our model parameters, and finally we give our evaluation results.

6.1 Datasets

We used the datasets given by Morpho Challenge 2009, and Cross Language Evaluation Forum (CLEF) to train our system on 3 different languages: English, German, and Turkish. For PoS clustering, we used the given corpora by Morpho Challenge 2009¹. For the clustering of the words in the word lists to be segmented we used the datasets supplied by CLEF organization². We used the CLEF dataset to obtain context distributions of the words in German, and in English. For Turkish, we used a collection manually collected newspaper archives.

6.2 Model Parameters

Our model is unsupervised, but it requires two prior parameters to be manually set. First, the threshold t on the conditional probability of the morpheme given its PoS cluster, $P(m|c)$, needs

¹<http://www.cis.hut.fi/morphochallenge2009>

²<http://www.clef-campaign.org/>. English datasets: Los Angeles Times 1994 (425 mb), Glasgow Herald 1995 (154 mb). German datasets: Frankfurter Rundschau 1994 (320 mb), Der Spiegel 1994/95 (63 mb), SDA German 1994 (144 mb), SDA German 1995 (141 mb)

to be fixed. We tested different values of this parameter for each language to find a suitable value through trail and error and we set $t = 0.1$. Thus, only morphemes, m , with $P(m|c) > 0.1$ were considered. Second, the threshold on the expected accuracy, T , of merging two paradigms ϕ_1, ϕ_2 given in Equation 4 needs to be set. Smaller values of this threshold leads to bigger paradigms with more stems, but it decreases the accuracy. Several experiments were performed to find its optimum value for different languages and a value of $T = 0.75$ was chosen. Both thresholds t and T once set were unchanged across all experiments reported in this paper.

6.3 Evaluation & Results

The system was evaluated in Competition 1 of Morpho Challenge 2009. Precision is calculated by sampling pairs of words with the same morpheme(s) from the system output and checking this against the gold standard. Recall is calculated in a similar way but this time pairs of words with the same morpheme(s) are sampled from the gold standard and checked against the system output. F-measure is the harmonic mean of the precision, and the recall:

$$F - measure = \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (5)$$

Evaluation results corresponding to the English language are given in Table 2.

Table 2: Evaluation results for English

Language	Precision	Recall	F-measure
English	58.52%	44.82%	50.76%

Evaluation results corresponding to the German language are given in Table 3. We conducted two different experiments for German. In the first experiment, we used only the compounding rules (see Section 5.3) for the German word list. Since German heavily consists of compounds, the results in Table 3 show that the compounding rules have high precision but low recall. In the second experiment, we used the unsupervised model developed in this paper.

Table 3: Evaluation results for German

Language	Precision	Recall	F-measure
German - compound	73.16%	15.27%	25.27%
German - normal	57.67%	42.67%	49.05%

Evaluation results for Turkish are given in Table 4. Two different experiments were conducted for Turkish. In the first experiment, a validity check was performed while splitting the word recursively to decide whether to split the word. The validity check simply checks the membership of the given word in the Turkish corpus. If the rest of the word after splitting one morpheme exists in the corpus, the validity condition is assumed to be met. In the second experiment, no validity check is performed. Instead the morpheme dictionary is used. The morpheme dictionary is constructed from the learnt morphological paradigms by extracting all the morphemes to create a dictionary. The word is split recursively from the rightmost end by matching these with the morphemes in the morpheme dictionary. Our experiments show that the precision gets higher when a validity check is done but the recall is reduced. Since the Turkish dataset does not include all the forms of every word, the validity check is not reliable leading to a lower recall.

7 Conclusion & Future Work

In this paper, we have developed a model for unsupervised morphology learning that exploits the PoS categories induced by an unsupervised PoS tagger. To our knowledge there has been very lim-

Table 4: Evaluation results for Turkish

Language	Precision	Recall	F-measure
Turkish (validity)	73.03%	8.89%	15.86%
Turkish (no validity)	41.39%	38.13%	39.70%

ited work on combined learning of syntactic categories and morphology. Our results demonstrate that it is meaningful to use the syntactic categorial information for morphology learning. One problem with the current approach is that it requires a large amount of corpus for PoS clustering. If a word does not have enough context information due to corpus size, it can not be clustered. The system then segments this by using just the morpheme dictionary. This in turn leads to inaccurate segmentations. The current system also requires manual setting of some thresholds. Furthermore, the system is very sensitive to these thresholds.

In the near future, we plan to address the above issues with the current model. In particular, we are interested in generative models for joint learning of morphology and PoS categories.

References

- [1] Stephan Bordag. Unsupervised and knowledge-free morpheme segmentation and analysis. In *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum (CLEF)*, pages 881–891, 2007.
- [2] Michael R. Brent. Minimal generative models: A middle ground between neurons and triggers. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, 1993.
- [3] Michael R. Brent, Sreerama K. Murthy, and Andrew Lundberg. Discovering morphemic suffixes a case study in MDL induction. In *Fifth International Workshop on AI and Statistics*, pages 264–271, 1995.
- [4] Alexander Clark and Issco Tim. Combining distributional and morphological information for part of speech induction. In *Proceedings of the 10th Annual Meeting of the European Association for Computational Linguistics (EACL)*, pages 59–66, 2003.
- [5] Alexander Clark. Inducing syntactic categories by context distribution clustering. In *The Fourth Conference on Natural Language Learning (CoNLL)*, pages 91–94, 2000.
- [6] Mathias Creutz. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, pages 280–287, 2003.
- [7] Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL workshop on Morphological and phonological learning*, pages 21–30, 2002.
- [8] Hervé Déjean and Basse Norm. Morphemes as necessary concept for structures discovery from untagged corpora, 1998.
- [9] John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001.
- [10] John Goldsmith. An algorithm for the unsupervised learning of morphology. In *Natural Language Engineering*, volume 12, pages 353–371, 2006.
- [11] Zellig Sabbettai Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

- [12] Yu Hu, I. Matveeva, J. Goldsmith and C. Sprague. Using morphology and syntax together in unsupervised learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 20–27, June, 2005.
- [13] Fred Karlsson. Finnish grammar. *WSOY, Juva*, 1983.
- [14] Christian Monson. *Paramor: From Paradigm Structure to Natural Language Morphology Induction*. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2008.
- [15] Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 67–72, 2000.
- [16] Matthew G. Snover, Gaja E. Jarosz and Michael R. Brent. Unsupervised learning of morphology using a novel directed search algorithm: taking the first step. In *Proceedings of the ACL workshop on Morphological and phonological learning*, 2002.