Unsupervised Morpheme Segmentation Through Numerical Weighting and Thresholding

Joy Mahapatra[*]
Technische Universität Darmstadt
Germany
joymahapatra90@gmail.com

Sudip Kumar Naskar
Jadavpur University
Kolkata
sudip.naskar@cse.jdvu.ac.in

Abstract

This paper presents an unsupervised model for morpheme segmentation of words collected from any raw textual corpus of a natural language. The model incorporates a numerical weighting scheme with thresholding technique for finding legitimate morphemes from a given input corpus. Kneedle algorithm is used as a thresholding technique for determining legitimacy of the morphemes. We ran our experiments on five languages – English, Finish, Turkish, German and Bengali, and the model performance is comparable to the state-of-the-art systems.

1 Introduction

Morpheme segmentation of words is an essential part of many linguistic and natural language processing applications. Appropriate morpheme segmentation helps to understand the hidden structure of a language's words and how new words can be built from the existing words. In morpheme segmentation, a word is divided into a stem morpheme and a single affix morpheme (for one-slot morphological languages) or multiple affix morphemes (for multi-slot morphological languages). Stem is also often referred to as base, root, lemma, etc., although they have subtle differences and are used in different contexts. An affix can be of many types; some of the most commonly understood affixes are prefix, suffix, infix, etc. However, for the proposed model, only prefixes and suffixes are considered. Two primary functional types of morphemes exist in morphology: inflectional and derivational morphemes. Inflectional morphemes are affixes that are used to create variant forms of a word in order to signal grammatical information; but they do not change the meaning of the word. Derivational morphemes are affixes that are used to derive new words with new meanings. Both types of morphemes are considered in our work.

The presented model's work principle falls into the category of *Unsupervised Learning of Morphology* (ULM) (Hammarström and Borin, 2011) which usually outputs a morphological structure description of a language from an input raw corpus of that language, provided that the system may need some semi-automatic or manual supervision. The objectives of an ULM based approach can vary. It generally ranges from demand for morphological description of a language to finding lexicon, paradigm list for stems, affix list, same-stem decision, inflectional table and much more. The objective of our proposed model is to discover the stem set and an affix set given a large corpus of a particular language.

Although there are many motivating factors behind ULM from both linguistic and practical point of view (Hammarström and Borin, 2011), the three major motivations are - providing a primary-step for language acquisition, reducing time-consuming manual effort in morphological analysis and language documentation. The first motivation is elicited from the necessity of grabbing primary details and learning basic word structures for a newly observed language. The second motivation is that unsupervised statistical approaches take less amount of time for accomplishing a task without taking much external efforts and resources. The third motivating factor is drawn from a linguistics point of view. It has been observed that in the current world, 80% of the world's languages (almost 7000 total languages) are spoken by only 100,000 speakers or less (Ostler, 2008). It has also been observed that many natural languages are at the verge of extinction (Krauss, 1992). Many linguists fear that with the extinction of such languages, many cultures and valuable information will be lost. They sug-

[*]Work done while at Jadavpur University.

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 298–304

gest taking help from any immediate quick procedures to restore those almost extinct language details (language documentation). A fast unsupervised approach for morpheme segmentation can provide an essential equipment for language documentation for such languages.

2 Related works

There exist many types of unsupervised morpheme segmentation models and ULM based systems with their own strengths and weaknesses. Hammarström and Borin (2011) classified the ULM models into four underlying types.

The first type emerged as border separation in words through substring frequency determination which explores the idea that if a substring occurs multiple times with other different substrings, then the former substring could be an affix morpheme, whereas the latter ones can be recognized as stem morphemes. After finding such substrings, this type of morphological analysis model tries to define the borders in words. The first-ever ULM based system (Harris, 1955) falls in this category of ULM which is a very popular ULM technique till date. Few researchers (Golcher, 2006; Hammarström, 2009) suggested morpheme segmentation using entropy.

The second type uses grouping and abstracting techniques and they first group all similar morphological words into a particular cluster among many existing ones, then find unique pattern for each cluster of words in such a way that the patterns can reveal all morphemes corresponding to the clusters. his approach is also very common and has multiple implementation examples (Schone, 2001; Yarowsky and Wicentowski, 2000; Wicentowski and Yarowsky, 2002; Wicentowski, 2004; Majumder et al., 2007).

The third ULM based approach (Mayfield and McNamee, 2003; De Pauw and Wagacha, 2007) is quite similar to basic machine learning based approaches. It first represents each word by multiple features and finally stems are separated from the affixes based on the feature values.

The last type of ULM technique is quite similar to the first ULM technique, with a small exception that prior to the border separation, words are categorized based on their phoneme structure (Rodrigues and Cavar, 2007). This ULM technique is applicable for non-concatenative morphology analysis, whereas the rest of the ULM techniques work mainly with concatenative morphological languages. Our proposed approach falls in the first category of ULM.

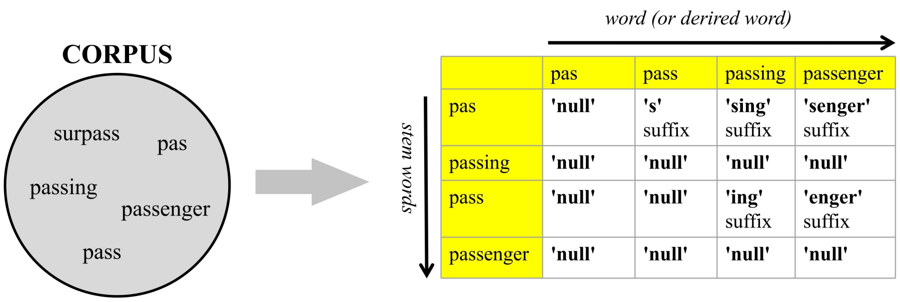
3 Proposed Method

The proposed morpheme segmentation model takes a raw, unannotated word dataset of an arbitrary language as input. Using a numerical weighting scheme with thresholding strategy, the model ultimately produces a set of stems and a set of affixes. The model also provides the morpheme segmentation of the words. It is to be noted that the model has been proposed and works well with concatenative, one-slot morphological languages (e.g., Bengali), although it is applicable to multi-slot morphological languages (e.g., Turkish, Finnish, etc.).

The proposed morpheme segmentation model for concatenative morphological languages has three basic modules. The first module is responsible for finding all probable initial morphemes (i.e., stems and affixes) from a raw text corpus. The second module scores the morphemes found by the first step. The third module finds out the optimal set of stems and affixes with unsupervised thresholding.

3.1 Morpheme Generation

This module finds out all probable stems and affixes by comparing every word with every other word in a text corpus. For example, by comparing the two words ‘pass’ and ‘passing’, one can easily perceive that ‘pass’ and ‘ing’ could be the stem and affix respectively. For an efficient storing and accessing mechanism of each stem, affix and stem-derived word (i.e., surface word), an implicit matrix ($\mathcal{M}$) type structure is considered, where the matrix columns represent stem-derived words and the rows represent the stems. Each element of the matrix represents an affix (i.e., a prefix or a suffix) or null, which when applied to the corresponding row-word, produces the corresponding column-word. A snapshot of the matrix is shown in Figure 1. Algorithm 1 outlines the process of generating the *stem-affix-word* matrix from the corpus words. To address the scalibility of this alrogithm, we have included a short discussion in Section 4.3.



	pas	pass	passing	passenger
pas	'null'	's' suffix	'sing' suffix	'senger' suffix
passing	'null'	'null'	'null'	'null'
pass	'null'	'null'	'ing' suffix	'enger' suffix
passenger	'null'	'null'	'null'	'null'

Figure 1: Representation of corpus as stem-affix-word matrix

```
Input: Raw text corpus $\mathcal{C}$ of language $\mathcal{L}$
Output: $\mathcal{M}$, two-dimentional matrix, with size
        $|\mathcal{C}| * |\mathcal{C}|$, where $|\mathcal{C}|$ is the number of unique
        words in $\mathcal{C}$ corpus
begin
    foreach element $(m)$ in matrix $\mathcal{M}$ do
        $m \leftarrow$ 'null'
    end
    foreach distinct word $w_1$ in $\mathcal{C}$ do
        foreach distinct word $w_2$ in $\mathcal{C}$ do
            if $w_2 = w_1 + a_1$, where $a_1$ is an affix
              then
                $\mathcal{M}[w_1][w_2] \leftarrow a_1$
                $\mathcal{M}[w_1][w_2].type \leftarrow$ 'suffix'
            else if $w_2 = a_1 + w_1$, where $a_1$ is an
              affix then
                $\mathcal{M}[w_1][w_2] \leftarrow a_1$
                $\mathcal{M}[w_1][w_2].type \leftarrow$ 'prefix'
            end
        end
    end
end
```

Algorithm 1: Generating all possible stems and affixes

3.2 Weighting Morphemes

We propose a weighting scheme that provides a ranking over the morphemes produced by Algorithm 1; the hypothesis is that higher ranked morphemes are likely to be legitimate morphemes of the language. The proposed weighting scheme works in three steps: independent scoring of the affixes, stem scoring through all its possible affixes, and joint stem-affix scoring.

3.2.1 Independent Affix Scoring

In this stage of the weighting scheme, every possible affix found in $\mathcal{M}$ is scored independently. If an affix works as both prefix and suffix, then two different scores are produced for that affix. The independent score for an affix is calculated from the number of different possible stems which appears adjacent to the affix. For an affix (a_x), we refer to this number as its branching factor (bf_{a_x}). Equation 1 shows the calculation of independent score (IS) of a_x from the branching factor of a_x.

$$IS(a_x) = \tanh \beta(bf_{a_x} - 1) \qquad (1)$$

This formulation of the affix score (as in Equation 1) was chosen for two major reasons. Firstly, affixes whose branching factor is 1 are canceled out since such affixes carry no or very little significance with regard to the legitimacy of the affix. Secondly, we want high independent score (close to 1) for all affixes above a certain value of branching factor so that affixes with very high branching factors can not dominate over affixes having low branching factors. Although the parameter β needs to be tuned for optimal performance, we chose a value of 2 for β for our experiments. Figure 2 shows the tangent hyperbolic function (cf. equation 1) for varying β values.

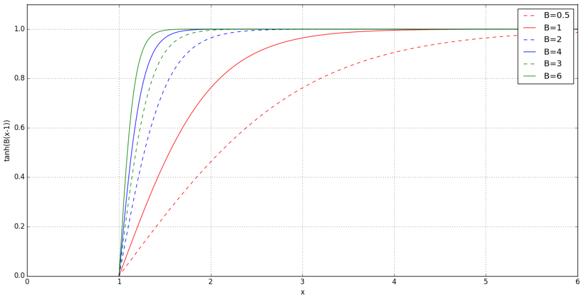


Figure 2: tanh(β(x-1)) function with varying β

Algorithm 1 only considers those stems which appear as words themselves in corpus $\mathcal{C}$. Algorithm 2 alleviates this shortcoming and modifies the matrix $\mathcal{M}$ to discover other possible legitimate stems (that do not appear as words in $\mathcal{C}$) with the help of independent affix scores. Algorithm 2 can also take care of morpheme segmentation in multislot morphological languages to some extent.

3.2.2 Affix-Dependent Stem Scoring

This stage determines the affix-dependent score (AdS) for each stem found by Algorithm 2. AdS

```
Input: Corpus $\mathcal{C}$ and Matrix $\mathcal{M}$ along with the
       corresponding independent affix scores and
       types
Output: $\mathcal{M}$, enriched with probable corpus-absent
        stems and adjusted for multi-slot morpheme
        segmentation
begin
    /* For multi-slot morphological
       language                                  */
    foreach $a_x$ such that $IS(a_x) == 0$ do
        Build a set of sets, $S^{a_x}$, where,
          $S^{a_x} = \{\{a_{x_1}, a_{x_2}, ..., a_{x_n}\} : a_x = concat(a_{x_1}.a_{x_2}...a_{x_n})$ and $\forall_i IS(a_{x_i}) \neq 0\}$;
        if $S^{a_x} \neq NULL$ then
            Define $mslot\_IS : s_k^{a_x} \in S^{a_x} \rightarrow \mathbb{N}$;
            $mslot\_IS(s_k^{a_x}) = \frac{\sum_{i=1}^{n}\{IS(a_{x_i}) : \text{where } a_{x_i} \in s_k^{a_x}\}}{cardinality(s_k^{a_x})}$;
            $Best_{S^{a_x}} \rightarrow \underset{s \in S^{a_x}}{argmax}\ mslot\_IS(s)$;
            $IS(a_x) \leftarrow mslot\_IS(Best_{S^{a_x}})$;
        end
    end
    /* Generating corpus-absent
       possible stems                            */
    foreach non-zero independent scored affix $a_x$ in
      $\mathcal{M}$ do
        foreach unique word $w_x$ in $\mathcal{C}$ do
            if $w_x = new_{stem} + a_x$ or
              $w_x = a_x + new_{stem}$ then
                if no row with $new_{stem}$ in $\mathcal{M}$ then
                    Make $\mathcal{M}[new_{stem}]$ row;
                    $\forall_i(\mathcal{M}[new_{stem}][i] \leftarrow null)$;
                end
                $\mathcal{M}[new_{stem}][w_x] \leftarrow$ '$a_x$';
                set $\mathcal{M}[new_{stem}][w_x].type$
                  accordingly
            end
        end
    end
end
```

Algorithm 2: Modifying $\mathcal{M}$ for multi-slot morphological languages and corpus-absent stems

is an indicator of the genuineness of a detected stem of being an actual stem. The AdS of a stem depends on its associated affixes in $\mathcal{M}$ and their independent scores. If a stem is associated with more zero independent scored affixes than non-zero independent scored affixes, then the stem loses its genuineness of being a valid stem. The more a stem is associated with non-zero independent scored affixes, the more reliable the stem is.

The AdS of $stem_x$ is computed as in Equation 2 where S is the sum of independent scores of affixes associated to $stem_x$, X and Y represent the number of non-zero and zero independent scored affixes, respectively, associated with $stem_x$, and $\alpha(\geq 1)$ is a penalty factor for associated zero independent scored affixes.

$$AdS(stem_x) = \frac{S}{X + \alpha.Y} \qquad (2)$$

Through adjusting the value of α, the affix-dependent score of a stem can be changed with the number of zero independent scored affixes. Large α value highly penalizes this score, whereas low α value do the opposite. For our experiments we fixed α as 2.

3.2.3 Joint Stem-Affix Scoring

$IS(a_x)$ determines the legitimacy of a_x of being an acutal affix. However, the linguistic authenticity of an affix is always estimated along a stem. For example, in English, the '*ing*' suffix holds a high independent score, but the chance of its association with the stem '*k*' (i.e., *k+ing*) is very low compared to the stem '*watch*' (i.e., *watch+ing*), for example. Therefore, a joint scoring mechanism taking into account both affix and stem is required.

The joint stem-affix score ($JSAS$) of $stem_x$ and a_y is computed as in Equation 3.

$$JSAS(stem_x, a_y) = AdS(stem_x) * IS(a_y) \qquad (3)$$

3.3 Finding Optimal Set of Stems and Affixes with Unsupervised Thresholding

This is the final operational stage of the proposed model which results in an optimal stem set and an affix set (i.e., paradigm list) from $\mathcal{M}$ based on the $JSAS$ scores. A threshold on $JSAS$ is required for achieving this. For the proposed model, a value of 2 was considered for both β and α. The threshold value ($Threshold_{JSAS}$) for $JSAS$ is determined using the Kneedle algorithm (Satopaa et al., 2011), an unsupervised approach for finding the knee points on curves. The knee points in a tunable system parameter's curve represent advantageous values for that parameter which balance the overall system performance compared to most of the other points in that curve. Unlike other knee points detection approaches, the Kneedle algorithm does not incorporate any system specific information to find out the knee points. This aspect of the Kneedle algorithm helps keep our model almost unsupervised.

3.4 Justification of Our Morpheme Weighting Scheme

Although intuitions behind deriving our morpheme weighting scheme may look like a heuristic procedure, actually, the weighting scheme is firmly rooted in basic linguistic postulates. We came up with those methods (equations) for the weighting scheme after attending a few conventional linguistic and mathematical rules. The Independent Affix Scoring (IS) method can be justified through the Zipf's empirical law. It has been observed for many years that most of the languages and even random texts follow the Zipf's law (Li, 1992). According to the empirical law, in a large dataset, for every individual word ($word$) the multiplication of its rank in the corpus (r_{word}) and count frequency of the word ($CountFreq_{word}$) remains the same (i.e., $r_{word} * CountFreq_{word} \equiv constant$). Figure 3 shows a sample distribution of the Zipf's law.

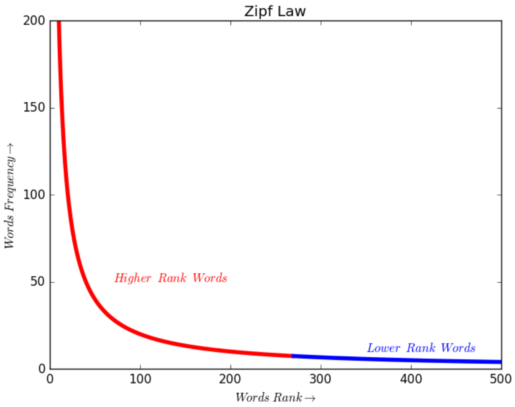


Figure 3: An Ideal Example of Zipf's Law

From this empirical law, it is not difficult to understand that most of the lower ranked words of a large corpus appear for a very few number of times (e,g,. only once or twice). It might so happen that these words with low empirical counts can also possess new affixes. Those affixes will also exist with an unquestionably low count. Therefore, we formulated Equation 1 to select all those low count affixes by assigning them identical weights as the high counted affixes. On the other hand, the second method of the morpheme weighting scheme, Affix-Dependent Stem Scoring (AdS), can be justified when it is seen as a regular mathematical normalization technique with adding denominator penalties for zero independent scored affixes (since zero independent scored affixes are really insignificant). The last method of the weighting scheme, Joint Stem-Affix Scoring ($JSAS$), is nothing but a single objective function comprised of IS and AdS as two distinct objectives.

4 Experiments

4.1 Datasets and Experimental Setup

The proposed method of morpheme segmentation was experimented on five languages – English, Bengali, Finnish, German and Turkish. For English, Turkish, German and Finnish, we used the Morpho-Challenge[1] datasets which provide both raw text corpora as well as gold-standard testsets. The gold-standard datasets mostly contain multi-slot morpheme segmentation samples. The datasets also come with evaluation results of a baseline system (Morfessor) (Creutz and Lagus, 2007). The Morpho-Challenge datasets' training data contains 617,297, 2,338,323, 2,928,030 and 878,036 distinct Turkish, German, Finnish and English words respectively. The test sets contain 1,000 words for each of those four languages. The Dataset also provides a perl script for evaluation on the gold-standard data. For Bengali, we used a gold standard testset (containing 14,034 words) developed in-house and collected a raw corpus (containing 28,927 unique words) by crawling an online Bengali newspaper. Unlike Morpho-Challenge dataset, the Bengali gold-standard data mostly contain single-slot morpheme segmentation examples. The output generated by the system heavily depends on choosing a proper threshold value for $JSAS$ which we determined using the Kneedle algorithm. Figure 4 graphically shows the $JSAS$ score thresholding by Kneedle algorithm for the Bengali dataset.

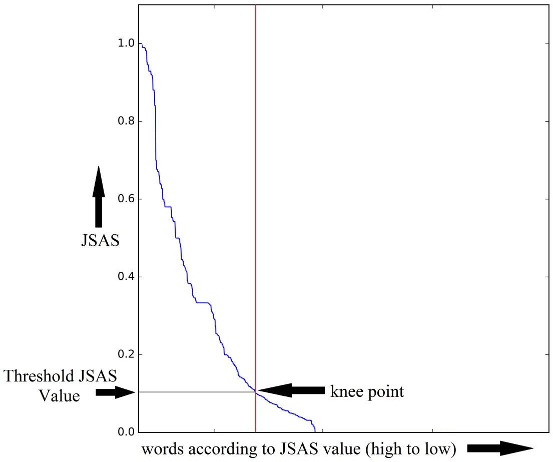


Figure 4: Thresholding using Kneedle algorithm

[1] http://morpho.aalto.fi/events/morphochallenge2010/

Table 1: Evaluation Results

Metric	System	Bengali	English	Turkish	Finnish	German
Precision	B	0.488	0.456	0.421	0.464	0.433
	MB	-	0.813	0.896	0.906	0.828
	P	0.853	0.763	0.695	0.612	0.746
Recall	B	0.842	0.713	0.627	0.675	0.630
	MB	-	0.417	0.177	0.143	0.197
	P	0.724	0.622	0.573	0.542	0.526
F-measure	B	0.617	0.556	0.504	0.550	0.513
	MB	-	0.551	0.296	0.248	0.319
	P	*0.783*	*0.685*	0.628	0.575	*0.617*
	Best	-	0.674	*0.653*	*0.625*	0.508

4.2 Evaluation

System performance was evaluated with precision, recall and f-measure (F1-measure) and the evaluation results are reported in Table 1. We developed a new baseline model which is similar to the proposed model except that it considers $IS_{baseline}(a_x) = bf_{a_x}$ instead of transforming the branching factor through hyperbolic tangent function. Table 1 presents the performance of the newly constructed baseline (*B*), Morfessor baseline (*MB*), the proposed model (*P*) and the best results (*Best*) reported so far on this dataset[2]. The baseline model produces high recall, however, due to absence of a proper thresholding mechanism, it results in low precision and hence low F-measure. We observed that the proposed model shows much better results for single-slot morpheme segmentation compared to multi-slot morpheme segmentation. With the aforementioned set-up, the best performance was observed for Bengali (F-measure 0.783) and the lowest for Finnish (F-measure 0.575). The proposed model outperformed the best results reported so far for English and German on this dataset. Considering that our model is almost unsupervised and it does not require any resources other than a vocabulary, our model results are, overall, comparable with the best results reported on this dataset obtained with semi-supervised approaches.

4.3 Scalabiliy

To keep our morpheme segmentation method scalable towards large vocabulary, we introduced multiple trie data structures to implement the implicit matrix structured shape for storing the stems and affixes. The trie implementation significantly reduces our system running time because of its efficient searching and storing mechanism compared to an ordinary two-dimensional array.

Our model took 1,624.28616 seconds for finding out all possible morpheme segmentations over all the datasets for the mentioned languages. We carried out the entire task on a computer with Intel Core2Duo processor and 4 gigabytes RAM.

5 Conclusions

In this paper we presented an almost unsupervised model for morpheme segmentation given a text corpus. The proposed model uses statistical scoring technique with an unsupervised thresholding algorithm. The model performs bettter on single-slot morpheme segmentation than multi-slot morpheme segmentation. The proposed model yields performance comparable to state-of-the-art performance and outperforms the best results reported so far on English and German.

Acknowledgments

We would like to thank the anonymous reviewers for their feedback. Sudip Kumar Naskar is supported by Media Lab Asia, MeitY, Government of India, under the Young Faculty Research Fellowship of the Visvesvaraya PhD Scheme for Electronics & IT.

References

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1), January.

Guy De Pauw and Peter Waiganjo Wagacha. 2007. Bootstrapping morphological analysis of gikuyu using unsupervised maximum entropy learning. In

[2] http://morpho.aalto.fi/events/morphochallenge2010/results/

Proceedings of the eighth INTERSPEECH conference. Citeseer.

Felix Golcher. 2006. Statistical text segmentation with partial structure analysis. *Proceedings of KONVENS 2006*, pages 44–51.

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

Harald Hammarström. 2009. *Unsupervised Learning of Morphology and the Languages of the World*. Ph.D. thesis, University of Gothenburg.

Zellig S Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Michael Krauss. 1992. The worlds languages in crisis. *Language*, 68(1):4–10.

Wentian Li. 1992. Random texts exhibit zipf's-law-like word frequency distribution. *IEEE Transactions on information theory*, 38(6):1842–1845.

Prasenjit Majumder, Mandar Mitra, and Dipasree Pal. 2007. Bulgarian, hungarian and czech stemming using yass. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 49–56. Springer.

James Mayfield and Paul McNamee. 2003. Single n-gram stemming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 415–416. ACM.

Nicholas Ostler. 2008. Is it globalization that endangers languages. UNESCO/UNU Conference: Globalization and Languages: Building our Rich Heritage.

Paul Rodrigues and Damir Cavar. 2007. Learning arabic morphology using statistical constraint-satisfaction models. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, 289:63.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 166–171. IEEE.

Patrick John Schone. 2001. *Toward knowledge-free induction of machine-readable dictionaries*. Ph.D. thesis, University of Colorado.

Richard Wicentowski and David Yarowsky. 2002. *Modeling and learning multilingual inflectional morphology in a minimally supervised framework*. Ph.D. thesis, Ph. D. Thesis. Johns Hopkins University, Baltimore, Maryland.

Richard Wicentowski. 2004. Multilingual noise-robust supervised morphological analysis using the wordframe model. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pages 70–77. Association for Computational Linguistics.

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*, pages 207–216. Association for Computational Linguistics.