Thesis for the Degree of Licentiate of Engineering

Unsupervised Learning of Morphology: Survey, Model, Algorithm and Experiments

Harald Hammarström

Department of Computing Science
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, October 2007

Unsupervised Learning of Morphology:
Survey, Model, Algorithm and Experiments
Harald Hammarström

Technical Report no. 46L
ISSN 1652-876X
Department of Computer Science and Engineering
Language Technology Research Group

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
Telephone + 46 (0)31-772 1000

Printed at Chalmers, Göteborg, Sweden, 2007

Sammanfattning (Svenska)

Denna avhandling innehåller arbete på ett specifikt problem inom området språkteknologi. Problemet kan beskrivas som följer:

> Kan en dator extrahera en beskrivning av hur ord böjs i ett naturligt språk om den bara har tillgång till skriven text i språket?

Problemet kallas ofta för Unsupervised Learning of Morphology (ULM) och har ett brett spektrum av applikationer inom språkteknologi, såsom datamaskinell översättning, dokumentkategorisering och informationssökning. ULM-problemet är också relevant för lingvistisk teori och kan bidra till att förbättra empiriska undersökningar inom delområdena kvantitativ lingvistik och lingvistisk typologi.

Den första delen av avhandlingen innehåller en kartläggning av forskning som gjorts på ULM-problemet. Alla större och mindre arbetsinriktningar omnämns och ges en mycket kort karakterisering. Olika angreppssätt som har varit dominerade inom området som helhet tas upp och diskuteras kritiskt. Den allmänna bilden som översikten ger är att mycket arbete har gjorts om åtskilliga gånger med relativt lite utbyte och utveckling av tekniker.

Den andra delen av avhandlingen beskriver en förenklad modell för konkatenativ affixering, dvs hur stammar och affix strängas ihop till ord. Modellen säger att ord består av högfrekventa strängar ("stammar") som sammanfogas med lågfrekventa strängar ("affix"), som till exempel engelskans *play-ing*. Sedan visas att från en mängd ord som konstruerats enligt modellen, så kan affixen extraheras med sin korrekta segmentering. Extraktionsalgoritmen utvärderas impressionistiskt på en mängd olika naturliga språk.

Affixextraktionsalgoritmen producerar inte en fullständig beskrivning av ett böjningssystem – den producerar bara en lista av affix. Den tredje och fjärde delen av denna avhandling visar hur den kan användas i vidare morfologisk analys.

I den tredje delen presenteras en algoritm som avgör om två givna ord är böjningar av samma stam. Nyckelidén är en mätmetod som kvantifierar vilka affix som tenderar att dyka upp på samma mängd stammar. Algoritmen är fri från parametrar och mänsklig inblandning, och fungerar lika bra för språk som har väldigt olika morfologisk typologi. Algoritmen har en nästan perfekt korrekthet på ordpar som tagna från löpande text.

I den fjärde delen utnyttjas affixextraktionen för språkidentifikationsproblemet, dvs att avgöra vilket språk en given text är skriven på. Moderna tekniker för språkidentifikation använder oftast en frekvenstabell över 3-gram som 'fingeravtryck' av ett språk. Visserligen är detta angreppssätt väldigt framgångsrikt i praktiken (typiskt 99%-ig korrekthet) om texten som ska språkbestämmas är i storleksordningen 100 tecken eller mer, men

samma teknik kan inte användas för att tillförlitligt språkbestämma kortare ingångstext och inte heller kan det avgöras om ingångstexten egentligen är en sammanfogning av text från flera olika språk. Därför introduceras en mer finkornig språkbestämningsmodell som siktar på att kunna klassificera ingångstext som inte behöver vara längre än ett ord. I princip gissas språket på ett ej tidigare sett ord på grundval av affix som förekommer på det (om det finns något). Många praktiska applikationer behöver inte denna finkorniga granularitetsnivå, men flerspråkig informationssökning är ett stort applikationsområde där ingångstexten oftast består av bara ett eller ett par ord. Algoritmen utsätts för en rigorös evaluering på en 32-språkig bibelparallellkorpus som visar konkurrenskraftiga resultat både på kort- och flerspråkig ingångstext, och dessutom inte bara på en mängd europeiska språk med liknande morfologisk typologi.

Abstract (English)

This thesis contains work on a specific problem in field of Language Technology. The problem can be described as follows:

> Can a computer extract a description of word conjugation in a natural language using only written text in the language?

The problem is often referred to as Unsupervised Learning of Morphology (ULM) and has a wide variety of Language Technology applications, including Machine Translation, Document Categorization and I nformation Retrieval. The ULM problem is also relevant for linguistic theory, and can serve to boost empirical investigations in subfields as Quantitative Linguistics and Linguistic Typology,

The first part of the thesis contains a comprehensive survey of work done on the ULM problem. All the minor and major lines of work are mentioned with a reference and a very brief characterization. Different approaches that have been prevalent in the field as a whole are highlighted and critically discussed. The general picture resulting from the survey is that much work has been repeated over and over, with little exchange and evolution of techniques.

The second part of the thesis describes a simple model of concatenative affixation, i.e., how stems and affixes are stringed together to form words. The model says that words consist of high-frequency strings ("affixes") attached to low-frequency strings ("stems"), e.g., as in the English *play-ing*. Then it is shown that from a set words constructed according to the model, the affixes can be extracted with their correct segmentation. The algorithm for extraction is impressionistically evaluated on a diverse set of natural languages.

The affix extraction algorithm does not output a full-fledged description of conjugational patterns – it only produces a list of affixes. The third and fourth parts of the thesis show how it can be used in further morphological analysis.

In the third part, an algorithm is presented that decides if two given words are conjugations of the same stem. The key part is the development of a metric for quantifying which endings tend to attach to the same set of stems. The algorithm has no parameters or human input and works equally well for languages with widely different morphological typology. It achieves almost perfect accuracy on word pairs selected from running text.

In the fourth part, the affix extraction model is exploited for the written language identification problem, i.e., to decide which natural language a given text is written in. Existing state-of-the-art techniques to identify the language of a written text most often use a 3-gram frequency table as basis for 'fingerprinting' a language. While this approach performs very well in practice (99%-ish accuracy) if the text to be classified is of size, say,

100 characters or more, it cannot be reliably used to classify even shorter input, nor can it detect if the input is a concatenation of text from several languages. Therefore a more fine-grained model is presented which aims at reliable classification of input as short as one word. In essence, the language of an unseen word is guessed based on any salient affixes that appear on it. Many practical applications do not need this fine level of granularity, but Multilingual Information Retrieval is a major target area where input is usually only one or a few words. The algorithm is given a rigorous evaluation on a 32-language parallel bible corpus showing competitive accuracy on short input as well as multi-lingual input, and not only for a set of European languages with similar morphological typology.

The four papers included in this thesis have been published previously as follows:

- **Paper I**:

  Hammarström, H. (2007b). A survey and classification of methods for (mostly) unsupervised learning of morphology. In *NODALIDA 2007, the 16th Nordic Conference of Computational Linguistics, Tartu, Estonia, 25-26 May 2007*. NEALT.

  > This paper has been expanded to include all references explicitly as opposed to the space-limited published version.

- **Paper II**:

  Hammarström, H. (2006a). A naive theory of morphology and an algorithm for extraction. In Wicentowski, R. and Kondrak, G., editors, *SIGPHON 2006: Eighth Meeting of the Proceedings of the ACL Special Interest Group on Computational Phonology, 8 June 2006, New York City, USA*, pages 79–88. Association for Computational Linguistics. `http://www.cs.chalmers.se/~harald2/sigphon06.pdf`.

- **Paper III**:

  Hammarström, H. (2006b). Poor man's stemming: Unsupervised recognition of same-stem words. In Ng, H. T., Leong, M.-K., Kan, M.-Y., and Ji, D., editors, *Information Retrieval Technology: Proceedings of the Third Asia Information retrieval Symposium, AIRS 2006, Singapore, October 2006*, volume 4182 of *Lecture Notes in Computer Science*, pages 323–337. Springer-Verlag, Berlin.

- **Paper IV**:

  Hammarström, H. (2007a). A fine-grained model for language identification. In *Proceedings of iNEWS-07 Workshop at SIGIR 2007, 23-27 July 2007, Amsterdam*, pages 14–20. ACM.

Acknowledgements

I would like to thank a number of people who, in various ways, were of importance for my writing this thesis. Of the people in the Computer Science Department I would especially like to thank my supervisor Bengt Nordström for constant support, my second supervisor Aarne Ranta for constant support, Devdatt Dubhashi for constant harassment, Markus Forsberg, Björn Bringert, Håkan Burden, Alejandro Russo, David Wahlstedt, Libertad Tansini and all other fellow PhD students at the department for contributing to a great research environment. Outside the department, I would also like to thank Jens Allwood, Lars Borin, Anju Saxena, John Löwenadler, Lilja Øvrelid, the Africanists Karsten Legère, Christina Thornell, Eva-Marie Ström, Malin Petzell, Helene Fatima Idris for providing refuge into the wonderful world of linguistics. Further back in time, I would like to thank my old classmates from Uppsala who were instrumental in getting me hooked on Computer Science in the first place: Tomas Fägerlind (who has also taught me everything I know about humans), Magnus Ågren, Olof Dahlberg, Jim Holmström, Lars "Ars" Göransson, Mattias Jakobsson, Per "upp över 100" Sahlin and the charming Jonas "Norris" Grönlund.

Contents

Chapter 1

Introduction

The work described in this thesis is in the area of Language Technology (LT), here defined as *the study of computer-aided processing of natural languages*. The ultimate goal of LT is to allow computers to deal with ("understand") natural language as humans do, which would make computers enormously more useful to humans. As of now, this goal is very far off, and we are happy if we can make progress on smaller subtasks, even if they do not achieve perfect accuracy. The problem studied in this thesis is one such subtask, and can be described as follows:

> Given a large collection of written text in a given natural language, can a computer, without any explicit knowledge about the language, extract a description of how words are conjugated in that language?

The problem is often referred to as Unsupervised Learning of Morphology, but also (Automatic) Induction of Morphology, Morpheme Discovery, Word Segmentation, Algorithmic Morphology, quantitative Morphsegmentierung (in German) and other variants have been used. Of these, Unsupervised Learning of Morphology (ULM) is fairly common and faces the least risk of misunderstanding, so it will be used throughout the present work.

In the Computer Science tradition, the solution to task such as this amounts to a) providing a formal description of the problem (in terms of sets, strings, logical conditions and the like) into which real-world instances are approximated, b) providing a step-by-step description of a method, i.e., an algorithm, to compute the desired output from the input and c) a proof or argument for the correctness and (if known) the optimality of the algorithm. Remarkably, in the 1940s, long before the Computer Science had matured as a field, and long before computers became practical to use, so-called structural linguists we asking for a solution of the exactly the same kind to the ULM and related problems, but from a different perspective. The interest was not so much putting computers to work as to learn how

linguistic analysis could be understood, which has particular implications for linguistic theory and possibly child language acquisition. As with most work in Language Technology, the present work will draw on experiences both Computer Science and Linguistics, and hopefully contribute to all.

The ULM problem is stated above in rather abstract terms. One might ask for specifics in terms of which languages are targeted, what (implicit) knowledge is allowed, how high accuracy is the aim, if there are speed requirements, how much text input is needed, what is meant by a description of conjugating words, is a black-box solution adequate or do we have to understand the inner workings, what is assumed about the written form of a language and so on. All these aspects with be elaborated on in the thesis. However, in essence, we target a much wider range of languages than English, but if the input language is the English New Testament[1] the desired output is any kind of description that tells us that forms like played and playing are conjugations of the same stem, and that see and sea aren't, perhaps reaching 90% accuracy on such pairs. No knowledge at all of forms is to be supplied but a small number of parameters and assumptions about suffix-length can be tolerated, whereas running time is not a priority.

Word-form analysis, or morphological analysis (see below), is generally the first step in computational analysis of natural language, and as such has a wide variety of LT applications, including Machine Translation, Document Categorization and Information Retrieval. ULM can also serve to boost investigations in Linguistics, especially the subfields Quantitative Linguistics and Linguistic Typology, and potentially contribute to linguistic theory.

A legitimate question is about the stipulation that distributional criteria alone should serve as the only source of knowledge for the computer. Why cannot a little or a lot of human knowledge about a language be hard-wired in order to describe how words are conjugated? This is indeed an option, and has been the way to handle the matter for virtually all languages committed to computational treatment, but it normally requires a lot of human effort. Roughly the amount of work of an MA thesis is needed to computationally implement conjugational patterns and an unspecified but huge amount of work to list legal lexical items.[2] Therefore, the ULM-problem as specified, has an important role to play. First, it would be a great benefit to rid us of the human effort of implementing conjugational patterns for the next range of languages to receive computational treatment. Second, even for languages which have this already, along with huge lists of lexical items, open domain texts will always contain a fair share of (inflected) previously unknown words, that are not in the lexicon (Forsberg et al. 2006). There has to be strategy for such out-of-dictionary words – a ULM-solving algorithm is

[1] 785066 tokens/running words versus 12999 unique words/types (King James 1977).

[2] Because of this, most such implementations have so far not been released to the public domain and have sometimes been kept in formats with poor portability, but there is in principle no reason why it should continue to be so, cf. Forsberg (2007).

one possibility. It could also turn out that the ULM-problem cannot, in some sense, be solved without explicit human-derived linguistic knowledge. If such a proof, or a convincing argument, is found this constitutes a resolution to the ULM-problem as good as one which proves the existence of an ULM-solving algorithm.

1.1 Morphology

Morphology, in linguistics, refers to the study of word forms. That is, whenever a set of words in a language is recognized as having a unit of form and a unit of meaning in common, such as the words playing and played, morphology tries to account for the way in which these forms may be constructed. Morphology as here defined presupposes a definition of 'word'. Throughout this thesis, orthographic word[3], i.e., a string separated by spaces, will be used as the definition of a word, since this is the only practical alternative. Languages written using orthographies which do not separate 'words' or any similar-sized chunks, such as Thai or Chinese, will not be considered.

Morphologically, all words are stipulated to consist of one or more morphemes. A morpheme is defined as 'a smallest meaning-bearing unit of a language'. It is important to note that a unit as used here does not have to be a continuous or segmental unit; it could be anything systematic enough to discern a meaning in it – a discontinuous element or something which occurs 'on top' of something else, e.g., stress or a vowel change pattern. Sometimes a scalar distinction is made between lexical morphemes on the one end, which are said to have lexical meaning but no grammatical function, and, on the other end of the scale, grammatical morphemes, which are said to have grammatical function but no lexical meaning. We think this characterization is problematic in various ways, and throughout the work we will use the tangible terms high-frequency ("grammatical") and low-frequency ("lexical") to cover much the same ground; the terms are given here only because they figure in further definitions (see below) and we wish to be consistent with existing linguistic literature.

Typical labels for various kinds of morphology are the following:

[3]Linguists suspect that orthographic words, whether in languages with a long or short writing tradition, are not wholly arbitrary divisions, though this has never really been clarified. A promising hypothesis is that orthographic words in the modern western writing tradition is correlated with the phonological word, i.e., a segmental unit of speech which is bound by some language-specific phonological process (e.g., stress, vowel harmony etc cf. Hall 1999), but it not established that all languages naturally have a suitable such phonological process (cf. Russell 1999). Another possible definition is the 'grammatical word', i.e., unit which has to be moved around together whenever moved around in the sentence. The interplay, the cross-linguistics situation and further considerations are discussed in Dixon and Aikhenvald (2002a) and contributions in the same volume (Dixon and Aikhenvald 2002b) and a historical summary is in Julien (2006).

Isolating: morphemes are not combined into words at all.

Concatenative: morphemes are combined by preposing and/or appending

Non-concatenative: the way morphemes are combined cannot be modelled by preposing and/or appending alone

Additionally, the following labels are in common use, sometimes referring to all of the morphology of a given language, and sometimes referring to subparts of a given language:

Derivational: morphology that changes part-of-speech (for example, derives a noun from a verb). In some usages, derivational morphology includes all morphology that does not serve a grammatical function, even if it does not change the part-of-speech (then a noun derived from a noun is an instance of derivational morphology). May be concatenative or non-concatenative.

> Example in Classical Arabic (Haywood and Nahmad 1962): *sakana* 'he lived' (verb) vs. *sa:kin-* 'inhabitant' (noun), where the s-k-n root common to both means 'live', and the interspersed vowel sequence makes an inflected verb in the former and a noun stem in the latter.

Inflectional: morphology where one or more grammatical functions is realised in only one morpheme. May be concatenative or non-concatenative.

> Example in Polish (Swan 2002): *student* 'student' vs. *studentom* 'to the students', where the *-om* suffix means both dative ('to') and plural. Both functions are motivated in the language (e.g. verbs conjugate differently according to singular/plural and some prepositions take the dative case regardless of singular/plural), and there is no way to identify a plural and dative parts of *-om*.

Agglutinative: morphology where more than one morpheme, each with its own grammatical function, is glued in sequence. Must be concatenative or nearly so. (But a language can have concatenative morphology without being agglutinative, for example, if there is only one affix position.)

> Example in Turkish (Underhill 1976): *evlerimde* 'in my houses', where *ev* 'house', *-ler* plural, *-im* 'my' and *-de* 'in' are identifiable as such in the language.

Compounding: words combined of more than one lexical morpheme (as opposed only one lexical morpheme combined with any number of grammatical morphemes). May be concatenative or non-concatenative (though I have yet to see a language with sophisticated non-concatenative compounding).

The techniques for morphological segmentation in this work are devoted only to concatenative morphology, though the survey covers the broader range of morphology. We will also make a stronger assumption, namely, that grammatical morphemes[4] should have fixed positions relative to each other in the word. This assumption may be formalized as follows:

> If two grammatical morphemes s_x and s_y occur in the same word, the order of the two is always the same (i.e., either s_x comes in a position ahead of s_y or the other way around).

To be extra clear, this neither implies that both morphemes must occur whenever one of them does nor that they have to be adjacent. The assumption is important but, fortunately, not very restrictive at all – it turns out that it is obeyed almost everywhere in almost every natural languages. (But there are unquestionable exceptions. Those known to me are Kagulu, where the order of a specific pair of prefixes can be interchanged (Petzell 2007) as well as Chintang, Bantawa and possibly other Kiranti languages where prefix ordering in general is very free (Rai 1985) (Bickel et al. 2007).)

1.2 Survey of Previous Work

A comprehensive overview of previous approaches to the ULM-problems preludes the proposed new model-algorithm and experimental applications.

A separate need was felt to go through all previous work systematically since previous work tends to be only sporadically accounted for in recent articles. To some extent this is excusable, because of the usual space limit in conference papers. On the other hand, a lack of familiarity with previous work is observable, with the consequence that the advance proceeds in parallel rather than in serial.

While all sufficiently related articles and monographs (known to the author), are cited, they are too many to discuss in detail. However, overviewing all work done a research question over time allows certain trends to be discerned. The main disturbing trend is that there are masses of figures and heuristics, but no model (or theory) that can explain its results has emerged – in spite of roughly a hundred experimental approaches with widely varying assumptions, scope and parameters. (However, this does not contradict

[4]Two morphemes are said to be the same if they have the same form and the same function/meaning. Thus, two morphemes with the same form but not the same function/meaning, are considered different.

that useful algorithms have been developed and that foreign mathematical techniques have found a new promising area of application.)

1.3 A Naive Model of Affixation

In this thesis we propose a formal model for affixation, that is, a formal characterization of how high-frequency strings attach to low-frequency strings. Intuitively, suffixes attach to very many different stems, but one stem does not attach to very many different suffixes. This property creates asymmetric junctions which can be exploited for segmentation. The following is a synoptic restatement of the formal definition of the model.

Assume we have two sets of random strings over some alphabet Σ:

- Bases $B = \{b_1, b_2, \ldots, b_m\}$
- Suffixes $S = \{s_1, s_2, \ldots, s_n\}$

Such that:

Arbitrary Character Assumption (ACA) Each character $c \in \Sigma$ should be equally likely in any word-position for any member of B or S.

Next, build a set of affixed words $W \subseteq \{bs|b \in B, s \in S\}$, that is, a large set whose members are concatenations of the form bs for $b \in B, s \in S$, such that:

Frequent Flyer Assumption (FFA) : The members of S are frequent. Formally: Given any $s \in S$: $f_W(s) >> f_W(x)$ for all x such that 1. $|x| = |s|$; and 2. not $x \triangleleft s'$ for all $s' \in S$).

- $s \triangleleft w$: s is a terminal segment of the word w i.e., there exists a (possibly empty) string x such that $w = xs$
- $f_W(s) = |\{w \in W|s \triangleleft w\}|$: the (suffix) frequency, i.e., the number of words in W with terminal segment s

While this is believed by the author to capture an essential property, the model is naive because it is oblivious to numerous other affixational considerations known to occur in the languages of the world. The point in formulating even a naive model such a this, is that it can explain its results. For the cases for which it does work, it allows understanding why they work. Scrutinizing the model and the cases for which it doesn't work properly, we can pinpoint where the problem lies and refine the model at that particular point.

1.4 An Algorithm for Extracting Affixes

The key question is, if words in natural languages are constructed as W as above, can we recover the segmentation? That is, can we find B and S, given only W? We give an efficient algorithm to partially decide this. To be more specific, we can compute a score Z_W such that $Z_W(x) > Z_W(y)$ if $x \in S$ and $y \notin S$. In general, the converse need not hold, i.e., if both $x, y \in S$, or both $x, y \notin S$, then it may still be that $Z_W(x) > Z_W(y)$. This is equivalent to constructing a ranked list of all possible segments, where the true members of S appear at the top, and somewhere down the list the junk, i.e., non-members of S, start appearing and fill up the rest of the list. Thus, it is not said *where* on the list the true-affixes/junk border begins, just that there is a consistent such border.

The idea in the algorithm is to characterize the true suffixes in terms of three properties: frequency, curve-drop and random adjustment. These properties are interpretable to humans and involve no constants or parameters for any language. There is a clear sense in which members of S "should" come out as having the three suggested properties, and non-members of S "shouldn't", but, unfortunately, there is no formal proof for this. A formal proof would have to be in terms of probability, and so far our naive attempts to finding a proof have branched out into very inelegant case-by-case branches. It seems more profitable to reformulate the model in some way that allows a simpler proof.

1.5 Applications

We present two applications of the above affix-recognition algorithm.

Same-stem decision: To decide whether two words are of the same stem is a leaner problem than giving the actual segmentation or characterize the exact set of endings a stem may take. We give an intuitive algorithm based on the above segmentation model[5] and a metric for quantifying which endings tend to attach to the same set of stems. There are no parametres or human input and works equally well for languages with widely different morphological typology. It achieves almost perfect accuracy on word pairs selected from running text, but the accuracy would be lower if word pairs were sampled uniformly from a list of forms in a paradigm. (The explanation is that word pairs taken from text tend to exhibit frequent endings, and they are easier, whereas if they were taken from a paradigm list we would get infrequent endings more often, and they are more difficult.)

[5]The paper,m as written at the time, actually uses an earlier version of the affix extraction algorithm, which was not theoretically mature, but the results are identical when the newer, current, version of the algorithm is plugged in in its place.

Language Identification: Existing state-of-the-art techniques to identify the language of a written text most often use a 3-gram frequency table as basis for 'fingerprinting' a language. While this approach performs very well in practice (99%-ish accuracy) if the text to be classified is of size, say, 100 characters or more, it cannot be used reliably to classify even shorter input, nor can it detect if the input is a concatenation of text from several languages. Based on the above segmentation model, we present a more fine-grained model which aims at reliable classification of input as short as one word. It is heavier than the classic classifiers in that it stores a large frequency dictionary as well as an affix table, but with significant gains in elegance since the classifier is entirely without any parameters or the like. Classifying a short input query in multilingual information retrieval is the target application for which the method was developed, but also tools such as spell-checkers will benefit from recognising occasional interspersed foreign words. It is also acknowledged that a lot of practical applications do not need this fine level of granularity, and thus remain largely unbenefited by the new model. We also contribute with a rigorous evaluation on a 32-language parallel bible corpus, showing competitive accuracy on short input as well as multi-lingual input, and not only for a set of European languages with similar morphological typology.

1.6 Future Work

At least four lines of future work suggest themselves immediately, which are commented on below:

Agglutinative Morphology: The model as formulated above, has only one affix slot. It is safe to say that most languages do not fit this simplistic view, and this has a clear negative impact on the results. The generalization of both the algorithm and the model to several affix slots is already well under way, and will be finished in the near future.

Non-concatentaive Morphology: Many languages exhibit non-concatenative elements in the morphology. At this time, that challenge is not addressed. The ideas used in the model and algorithm do not generalize to non-catenative morphology because a key feature is the comparison between segments increased step-by-step in size. There are only quadratically many segments in the length of the word, but enumerating all increasing non-concatenative subunits of a word would be exponential and thus prohibitive. In order to by-pass the exponentially many potential sub-units, it seems that some pre-processing step is needed which cuts down on the possible non-concatenative shapes relevant in the language. At this time I have no clear idea of such an

algorithm, so there are no prospects for a solution in the near future.

Paradigms: In the end, the desired goal is more than a ranked list of suffixes, or a stem-stem decision; rather we want a complete segmentation of a given word and/or an abstract description of the conjugational patterns in the given language. The author believes that the latter, an abstract description of the conjugational patterns is the key to the former, complete segmentation, because it enables a great number of otherwise spurious segmentations to be cancelled. That is, we don't want any string that ends in a common ending, e.g., ring to be segmented according to the common ending, especially when there is no *red or *rs. The author has already worked on a specific way to group endings into groups that 'occur on the same stems' ("paradigms") (cf. Hammarström 2005) but the finalization of this work has been delayed by a systematic solution to the agglutination-matter (as above). It will be continued as soon as possible.

Part-of-Speech Induction: The next level after intra-word analysis is the analysis of which words may occur in which context. Contemporary linguistics uses the concept of word-class or part-of-speech to generalize on which words may occur adjacent to other words (e.g., in English, only a noun or adjective may follow the definite article). It will be interesting to investigate whether this problem too, is partly solvable in an unsupervised manner. It also opens the potential for morphemes to be labeled as function, which is wholly out-of-reach without recourse to context. A literature survey and some preliminary experiments have been carried out, but focus on part-of-speech induction will await a more mature state of the concatenative morphology analysis (as above).

Formal Proof: A proof, that guarantees a high probability of success given the model assumptions, would be vary valuable. It is possible that it is worth the effort to find someone to help me look deeper into probability theory.

Other possible continuations is to attempt to model sandhi phenomena, i.e., changes at the junction of two morphemes, which violate the concatenation postulate yet are common in natural languages.

1.7 Contributions

All the work in the present thesis is the sole and original work of the author, though it has benefitted much from discussions with Bengt Nordström and other people (see acknowledgments).

Chapter 2

Paper I: A Survey and Classification of Methods for (Mostly) Unsupervised Learning of Morphology

Harald Hammarström
Department of Computing Science
Chalmers University of Technology
harald2@cs.chalmers.se

Abstract

This paper surveys work on unsupervised learning of morphology. A fairly broad demarcation of the area is given, and a hierarchy of subgoals is established in order to properly characterize each line of work. All the minor and major lines of work are mentioned with a reference and a brief characterization. Different approaches that have been prevalent in the field as a whole are highlighted and critically discussed. The general picture resulting from the survey is that much work has been repeated over and over, with little exchange and evolution of techniques. All in all, the contribution of this paper is a very brief but comprehensive umbrellic synopsis to the research area.

2.1 Introduction

The problem of (mostly) unsupervised learning of morphology (ULM) may be broadly delineated as follows:

Input: Raw (unannotated) natural language text data

Output: A description of the morphological structure (there are various levels to be distinguished; see below) of the language of the input text

With: As little supervision, i.e., parameters, annotated bootstrapping data, model selection during development etc., as possible

Some approaches have explicit or implicit biases towards certain kinds of languages; they are nevertheless considered to be ULM for this survey.

Morphology may be narrowly taken as to include only derivational and grammatical affixation, where the number of affixations a root may take is finite and the order of affixation may not be permuted. This survey also subsumes attempts that take a broader view including clitics and compounding (and there seems to be no reasons in principle to exclude incorporation and lexical affixes). A lot of, but not all, approaches focus on concatenative morphology/compounding only.

All works in this survey operate on orthographic words – excluding word-segmentation for languages that do not mark word-boundaries orthographically.

One of the matters that varies the most between different authors is the desired outcome. It is useful to set up the implicational hierarchy shown in Table 2.1 (which need of course not correspond to steps taken in an actual algorithm). The division is implicational in the sense that if one can do the morphological analysis of a lower level in the table, one can also easily produce the analysis of any of the above levels. For example, if one can perform analysis into stem and affixes, one can decide if two word are of the same stem. The converse need not hold, it is perfectly possible to answer the question of whether two words are of the same stem with high accuracy, without having to commit what the actual stem is.

A lot of recent articles do not deal properly with previous and related work, some reinvent heuristics that have been sighted earlier, and there is little modularization taking place. Thus the time is ripe, even overdue, for a survey and classification of ideas in this area.

Our full bibliography of ULM-work comprises at least 100 articles/books (more if the level of unsupervised-ness is relaxed out of control) spanning from 1955 to 2006. Clearly, each article cannot be cited or discussed in detail, but we will cover each distinct line of work.

2.2 Roadmap and Synopsis of Earlier Studies

For reasons of space, very short characterizations of selected representatives of each line of work is given in Table 2.2. In addition, there is relevant work (Manning 1998; Borin 1991; Neuvel and Fulop 2002) on formalizing morphological regularities but which do not suggest an algorithm that performs on raw text data input. Furthermore, Zweigenbaum et al. (2003) and

Affix list	A list of the affixes.
↑	
Same-stem decision	Given two words, decide if they are affixations of the same stem.
↑	
Analysis	Given a word, analyze it into stem and affix(es).
↑	
Paradigm list	A list of the paradigms.
↑	
Lexicon+Paradigm	A list of the paradigms and a list of all stems with information of which paradigm each stem belongs to.

Table 2.1: Levels of power of morphological analysis. We do not make a distinction between probabilistic and non-probabilistic versions.

work referenced therein specifically targets medical-chemical vocabulary and Karagol-Ayan et al. (2006) specifically targets noisy data.

There is a slight tendency towards modularization as a fair amount of recent work focusses on finding paradigms, taking suitable stems-affix divisions as given (Goldsmith and O'Brien 2007; Chan 2006; Monson 2004; Monson et al. 2004; Bernhard 2007).

It was impossible to characterize methods and ideas in brief for each line of work because of the amount of detail necessary to give a relevant comparative picture. However, all work uses some kind of frequency count of n-character grams, and almost all trace their inspiration back to Harris (1955). In addition, some recent approches use a Minimum Description Length (MDL)-inspired formula as an optimization criterion of a given model. All the approaches to non-concatenative morphology involve an alignment-step, except Xanthos (2007) which rather attempts to learn phonological categories.

A few lines of work have tried to exploit other kinds of clues than character sequences, such as similarities in semantics or syntax between words (also acquired in a semi-supervised manner). A fair comparison of previous work in terms of accuracy figures is entirely impossible, not only because of the great variation in goals but also because most descriptions do not specify their algorithm(s) in enough detail. This aspect is better handled in controlled competitions, such as the Unsupervised Morpheme Analysis – Morpho Challenge 2007[1] which a task of segmentation of Finnish, English,

[1]Website `http://www.cis.hut.fi/morphochallenge2007/` accessed 10 January 2007.

	Model	Supervision	Experimentation	Learns what?
Harris (1955)+	C	T	English	Analysis
Andreev (1965)*	C	T	E-type (I)	Unclear
Gammon (1969)	C	T	E-type	Analysis
Lehmann (1973)	C	T	German	Analysis
Hafer and Weiss (1974)	C	T	English	Analysis
de Kock and Bossaert (1978)+	C	T	French/Spanish	Analysis
Klenk (1992)+	C	T	E-type	Analysis
Wothke and Schmidt (1992)+	C	T	German	Analysis
Klenk (1994)	NC	T	Arabic + E-type	Analysis
Langer (1991)	C	T	German	Analysis
Flenner (1995)+	C	T	Spanish	Analysis
Brent et al. (1995)	C	T	English	Analysis
Džeroski and Erjavec (2000)+	C	T	Slovene	Analysis
Kazakov and Manandhar (2001)+	C	T	French/English	Transducer
Gaussier (1999)	C	T + AP	English (I)	Paradigms
Lepage (1998)	NC	AP	Chinese to Arabic	New word
Goldsmith (2006)+	C	T	E-type (I)	Paradigms+Lexicon
Baroni (2003)+	C	T	E-type	Analysis
Clark (2001b)+	NC	# states	German/Arabic/English	Transducer
Déjean (1998a)+	C	T	E-type	Analysis
Schone (2001)+	C	T	E-type	Related pairs of words
Neuvel and Fulop (2002)	C	T	E-type (I)	Related pairs of words
Jacquemin (1997)	C	T	E-type	Related pairs of words
Sharma et al. (2002)	C	T	Assamese	Paradigms+lexicon
Baroni et al. (2002)	NC	T	English/German (I)	Ranked list of related word pairs
Creutz (2006)+	C	T	Finnish/Turkish/English	Analysis
Kontorovich et al. (2003)	C	T	English	Analysis
Snover and Brent (2003)+	C	T	English/Polish	Related pairs of words
Johnson and Martin (2003)	C	T	Inuktitut	Unclear
Wicentowski (2004)+	NC	AP	30-ish E-type	Transducers
Gelbukh et al. (2004)+	C	T	E-type	Analysis
Ćavar et al. (2004)+	C	T	Unclear	Paradigms
Argamon et al. (2004)	C	T	English	Analysis
Goldsmith et al. (2005)+	NC	T	Unclear	Unclear
Bacchin et al. (2005)+	C	T	E-type	Stemming
Oliver (2004, Ch. 4-5)	C	T	Catalan	Paradigms
Bordag (2005)	C	T	English/German	Analysis
Hathout (2005)	C	AP	English/French	Analysis
Kurimo et al. (2005)*	C	T	Finnish/Turkish/English	Analysis
Medina-Urrea (2006)+	C	T	Chuj/Ralámuri/Spanish	Analysis
Hammarström (2006b)+	C	-	Maori to Warlpiri	Same-stem
Arabsorkhi and Shamsfard (2006)	C	T	Persian	Analysis
Monson et al. (2007)	C	T	English/German	Paradigms+Lexicon
Demberg (2007)	NC	T	E-type	Analysis
Dasgupta and Ng (2007)	C	T	Finnish/Turkish/English	Analysis
Bernhard (2006)+	C	T	Finnish/Turkish/English	Analysis+Related sets of words
Xanthos (2007)+	NC	T	Arabic	Paradigms+Lexicon

Table 2.2: Very brief roadmap of earlier studies. Abbrevations in the Table: C = Concatenative, NC = Also non-concatenative, T = Thresholds and Parameters to be set by a human, AP = Aligned pairs of words, E-type = European Indo-European type languages, I = Impressionistic evaluation. * = this citation covers work by several different authors. + = entry also covers earlier work by the same author(s); see appendix. In concatenative morphology analysis = segmentation.

German and Turkish.

2.3 Discussion

Although the heuristic of Harris has had some success it was shown (in various interpretations) as early as Hafer and Weiss (1974) that it is not really sound – even for English. In the 2000s, probably independently, a slightly better extension of the same idea emerged, namely, to compile a set of words into a *trie* and predict boundaries at nodes with high actitivity, but this is not sound either as non-morphemic short common character sequences also show significant branching.

So far, all the approaches with mixed MDL-optimization are unsatisfactory on two main accounts; on the theoretical side, they still owe an explanation of why compression or MDL-inspired weighting schemes should give birth to segmentations coinciding with morphemes as linguists conceive of morphemes. On the experimental side, thresholds, supervised/developed parametres and selective input still cloud the success of reported results. What is clear, however, apart from whether it is theoretically motivated, is that MDL approaches are *useful*.

2.4 Conclusion

What emerges from the last 10 years of intensive research is that, essentially, different people have been doing the same thing with little exchange between each other.

2.5 Appendix

Ćavar et al. 2004+: See also Ćavar et al. 2005.

Creutz 2006+: See also Creutz and Lagus 2006; Creutz 2003; Creutz and Lagus 2005b: 2002; Hirsimäki et al. pear: 2005; Creutz et al. 2005a; Creutz and Lindén 2004; Creutz et al. 2005b; Creutz and Lagus 2005a; Hirsimäki et al. 2003; Creutz and Lagus 2004.

de Kock and Bossaert 1978+: See also de Kock and Bossaert 1974.

Klenk 1992+: See also Klenk 1985a: 1991; Klenk and Langer 1989; Klenk 1985b; Janßen 1992.

Wothke and Schmidt 1992+: See also Wothke 1984; Thurmair 1986.

Kazakov and Manandhar 2001+: See also Kazakov and Manandhar 1998; Kazakov 2000: 1997.

Déjean 1998a+: See also Déjean 1998b.

Flenner 1995+: See also Flenner 1992: 1994.

Džeroski and Erjavec 2000+: See also Džeroski and Erjavec 1997; Manandhar et al. 1998; Erjavec and Džeroski 2004.

Harris 1955+: See also Harris 1970.

Goldsmith 2006+: See also Goldsmith 2000: 2004; Hu et al. 2005b; Belkin and Goldsmith 2002; Goldsmith et al. 2001; Goldsmith and Hu 2004; Hu et al. 2005a; Goldsmith 2001.

Clark 2001b+: See also Clark 2002: 2001a.

Baroni 2003+: See also Baroni 2000.

Schone 2001+: See also Schone and Jurafsky 2000: 2001.

Bacchin et al. 2005+: See also Bacchin et al. 2002; Nunzio et al. 2004.

Medina-Urrea 2006+: See also Medina Urrea 2000; Medina Urrea and Díaz 2003; Medina Urrea 2006: 2003.

Snover and Brent 2003+: See also Snover and Brent 2001; Snover 2002; Snover et al. 2002; Brent 1999.

Wicentowski 2004+: See also Yarowsky and Wicentowski 2000; Wicentowski 2002.

Gelbukh et al. 2004+: See also Gelbukh and Sidorov 2003.

Xanthos 2007+: See also Xanthos et al. 2006.

Hammarström 2006b+: See also Hammarström 2007: 2006a: 2005.

Bernhard 2006+: See also Bernhard 2005.

Chapter 3

Paper II: A Naive Theory of Affixation and an Algorithm for Extraction

Harald Hammarström
Department of Computing Science
Chalmers University of Technology
harald2@cs.chalmers.se

Abstract

We present a novel approach to the unsupervised detection of affixes, that is, to extract a set of salient prefixes and suffixes from an unlabelled corpus of a language. The underlying theory makes no assumptions on whether the language uses a lot of morphology or not, whether it is prefixing or suffixing, or whether affixes are long or short. It does however make the assumption that 1. salient affixes have to be frequent, i.e., occur much more often that random segments of the same length, and that 2. words essentially are variable length sequences of random characters, e.g., a character should not occur in far too many words than random without a reason, such as being part of a very frequent affix. The affix extraction algorithm uses only information from fluctuation of frequencies, runs in linear time, and is free from thresholds and untransparent iterations. We demonstrate the usefulness of the approach with example case studies on typologically distant languages.

3.1 Introduction

The problem at hand can be described as follows:

Input : An unlabelled corpus of an arbitrary natural language

Output : A (possibly ranked) set of prefixes and suffixes corresponding to true prefixes and suffixes in the linguistic sense, i.e., well-segmented and with grammatical meaning, for the language in question.

Restrictions : We consider only concatenative morphology and assume that the corpus comes already segmented on the word level.

The theory and practice of the problem is relevant or even essential in fields such as child language acquisition, information retrieval and, of course, the fuller scope of computational morphology and its further layers of application (e.g., Machine Translation).

The reasons for attacking this problem in an unsupervised manner include advantages in elegance, economy of time and money (no annotated resources required), and the fact that the same technology may be used on new languages.

An outline of the paper is as follows: we start with some notation and basic definitions, with which we describe the theory that is intended to model the essential behaviour of affixation in natural languages. Then we describe in detail and with examples the thinking behind the affix extraction algorithm, which actually requires only a few lines to define mathematically. Next, we present and discuss some experimental results on typologically different languages. The paper then finishes with a brief but comprehensive characterization of related work and its differences to our work. At the very end we state the most important conclusions and ideas on future components of unsupervised morphological analysis.

3.2 A Naive Theory of Affixation

Notation and definitions:

- $w, s, b, x, y, \ldots \in \Sigma^*$: lowercase-letter variables range over strings of some alphabet Σ and are variously called words, segments, strings, etc.
- $s \triangleleft w$: s is a terminal segment of the word w i.e., there exists a (possibly empty) string x such that $w = xs$
- $W, S, \ldots \subseteq \Sigma^*$: capital-letter variables range over sets of words/strings/segments
- $f_W(s) = |\{w \in W | s \triangleleft w\}|$: the (suffix) frequency, i.e., the number of words in W with terminal segment s
- $S_W = \{s | s \triangleleft w \in W\}$: all terminal segments of the wordsin W

- $uf_W(u) = |\{(x, y)|xuy = w \in W\}|$: the substring frequency of u, i.e., the number times u occurs as a substring in the set of words W (x and y may be empty).
- $nf_W(u) = uf_W(u) - f_W(u)$: the non-final frequency of u, i.e. the substring frequency minus those in which it occurs as a suffix.
- $|\cdot|$: is overloaded to denote both the length of a string and the cardinality of a set

Assume we have two sets of random strings over some alphabet Σ:

- Bases $B = \{b_1, b_2, \ldots, b_m\}$
- Suffixes $S = \{s_1, s_2, \ldots, s_n\}$

Such that:

Arbitrary Character Assumption (ACA) Each character $c \in \Sigma$ should be equally likely in any word-position for any member of B or S.

Note that B and S need not be of the same cardinality and that any string, including the empty string, could end up belonging to both B and S. They need neither to be sampled from the same distribution; pace the requirement, the distributions from which B and S are drawn may differ in how much probability mass is given to strings of different lengths. For instance, it would not be violation if B were drawn from a a distribution favouring strings of length, say, 42 and S from a distribution with a strong bias for short strings.

Next, build a set of affixed words $W \subseteq \{bs|b \in B, s \in S\}$, that is, a large set whose members are concatenations of the form bs for $b \in B, s \in S$, such that:

Frequent Flyer Assumption (FFA) : The members of S are frequent. Formally: Given any $s \in S$: $f_W(s) >> f_W(x)$ for all x such that 1. $|x| = |s|$; and 2. not $x \lhd s'$ for all $s' \in S$).

In other words, if we call $s \in S$ a *true suffix* and we call x an *arbitrary segment* if it neither a true suffix nor the terminal segment of a true suffix, then any true suffix should have much higher frequency than an arbitrary segment of the same length.

One may legitimately ask to what extent words of real natural languages fit the construction model of W, with the strong ACA and FFA assumptions, outlined above. For instance, even though natural languages often aren't written phonemically, it is not hard to come up with languages that have phonotactic constraints on what may appear at the beginning or end of a word, e.g, Spanish **st-* may not begin a word and yields *est-* instead.

Positions	Distance
$\|\|p_1 - p_2\|\|$	0.47
$\|\|p_1 - p_3\|\|$	0.36
$\|\|p_1 - p_4\|\|$	0.37
$\|\|p_2 - p_3\|\|$	0.34
$\|\|p_2 - p_4\|\|$	0.23
$\|\|p_3 - p_4\|\|$	0.18

Table 3.1: Difference between character distributions according to word position.

Another violation of ACA is that (presumably all (Ladefoged 2005)) languages disallow or disprefer a consonant vs. a vowel conditioned by the vowel/consonant status of its predecessor. However, if a certain element occurs with *less* frequency than random (the best example would be click consonants which, in some languages e.g., Eastern !Xõo (Traill 1994), occur only initially), this will not be a practical problem.

As for FFA, we may have breaches such as Biblical Aramaic (Rosenthal 1995) where an old -$\bar{a}$ element appears on virtually everywhere on nouns, making it very frequent, but no longer has any synchronic meaning. Also, one can doubt the requirement that an affix should need to be frequent; for instance, the Classical Greek inflectional (lacking synchronic internal segmentation) alternative medial 3p. pl. aorist imperative ending -$\sigma\theta\omega\nu$ (Blomqvist and Jastrup 1998), is not common at all.

Just how realistic the assumptions are is an empirical question, whose answer must be judged by experiments on the relevant languages. In the absense of fully annotated test sets for diverse languages, and since the author does not have access to the Hutmegs/CELEX gold standard sets for Finnish and English (Creutz and Lindén 2004), we can only give some guidelining experimental data.

ACA On a New Testament corpus of Basque (Leizarraga 1571) we computed the probability of a character appearing in the initial, second, third or fourth position of the word. Since Basque is entirely suffixing, if it complied to ACA, we'd expect the distributions to be similar. However, if we look at the difference of the distributions in terms of variation distance between two probability distributions ($||p - q|| = \frac{1}{2}\sum_x |p(x) - q(x)|$), it shows that they differ considerably – especially the initial position proves more special (see table 3.1).

FFA As for the FFA, we checked a corpus of bible portions of Warlpiri (Summer Institute of Linguistics 2001). This was chosen because it is one of the few languages known to the author where data was available and which has a decent amount of frequent suffixes which are also long, e.g., case affixes are typically bisyllabic phonologically and five-ish

characters long orthographically. Since the orthography used marked segmentation, it was easy to compute FFA statistics on the words as removed from segmentation marking. Comparing with the lists in Nash (1980, Ch. 2) it turns out that FFA is remarkably stable for all grammatical suffixes occurring in the outermost layer. There are however the expected kind of breaches; e.g., a tense suffix *-ku* combined with a last vowel *-u* which is frequent in some frequent preceding affixes making the terminal segment *-uku* more frequent than some genuine three-letter suffixes.

The language known to the author which has shown the most systematic disconcord with the FFA is Haitian Creole (also in bible corpus experiments (American Bible Society 1999)). Haitian creole has very little morphology of its own but owes the lion's share of it's words from French. French derivational morphemes abound in these words, e.g., *-syon*, which have been carefully shown by (Lefebvre 2004) not to be productive in Haitian Creole. Thus, the little morphology there is in Haitian creole is very difficult to get at without also getting the French relics.

3.3 An Algorithm for Affix Extraction

The key question is, if words in natural languages are constructed as W explained above, can we recover the segmentation? That is, can we find B and S, given only W? The answer is yes, we can partially decide this. To be more specific, we can compute a score Z_W such that $Z_W(x) > Z_W(y)$ if $x \in S$ and $y \notin S$. In general, the converse need not hold, i.e., if both $x, y \in S$, or both $x, y \notin S$, then it may still be that $Z_W(x) > Z_W(y)$. This is equivalent to constructing a ranked list of all possible segments, where the true members of S appear at the top, and somewhere down the list the junk, i.e., non-members of S, start appearing and fill up the rest of the list. Thus, it is not said *where* on the list the true-affixes/junk border begins, just that there is a consistent such border.

Now, how should this list be computed? All terminal segments are contained in the set S_W, the question is just to order them. We shall now define three properties that we argue will be enough to put the S-belonging affixes at the top. For a terminal segment s, define:

Frequency The frequency $f_W(s)$ of s (as a terminal segment).

Curve Drop First, for s, define its curve $C_s(c)$ which is a probability distribution on Σ:

$$C_s(c) = \frac{f_W(cs)}{f_W(s)}$$

Next, more importantly, define its *curve drop* $\overline{C}(s)$ which is a value in $[0, 1]$:

$$\overline{C}(s) = \frac{1 - max_c(C_s(c))}{1 - \frac{1}{|\Sigma|}}$$

Random Adjustment First, for s, define its probability as:

$$P_W(s) = \frac{f_W(s)}{\sum_{s' \in S_W} f_W(s')}$$

Second, equally straightfowardly, for an arbitrary segment u, define its non-final probability as:

$$nP_W(u) = \frac{nf_W(u)}{\sum_{u'} nf_W(u')}$$

Finally, for a terminal segment s, define its *random adjustment* $RA(s)$ which a value in Q^+:

$$RA(s) = \begin{cases} \frac{P_W(s)}{nP_W(s)} & \text{if } nP_W(s) > 0 \\ 1.0 & \text{otherwise} \end{cases}$$

It is appropriate now to show the intuition behind the definitions. There isn't much to comment on frequency, so we'll go to curve drop and random adjustment. All examples in this section come from the Brown corpus (Francis and Kucera 1964) of one million tokens ($|W| = 47178$ and $|S_W| = 154407$).

The curve drop measure is meant to predict when a suffix is well-segmented to the left. Consider a suffix s, in all the words on which it appears, there is a preceding character c. Figure 3.1 shows examples of the frequency distribution on preceding character for example suffixes *-ing* and *-ng*. The reasoning is as follows. If s is a true suffix and is well-segmented to the left, then its curve-drop value should be high. Frequent true suffixes that attach to bases whose last character is random should have a close to uniform curve. On the other hand, if the curve drop value is low it means there is a character that suspiciously often precedes s. However, if s weren't a true suffix to begin with, perhaps just a frequent but random character, then we expect it's curve drop value to be high too! To exemplify this, we have $\overline{C}(ing) \approx 0.833$, $\overline{C}(ng) \approx 0.029$ and $\overline{C}(a) \approx 0.851$.

The random adjustment measure it precisely to distinguish what a "frequent but random segment" is, that is, discriminate e.g *-a* versus *-ing* as well as *-a* versus *-ng*. Now, how does one know whether something is random or not? One approach would be to say the shorter the segment the more random. Although it's possible to get this to work reasonably well in practice,

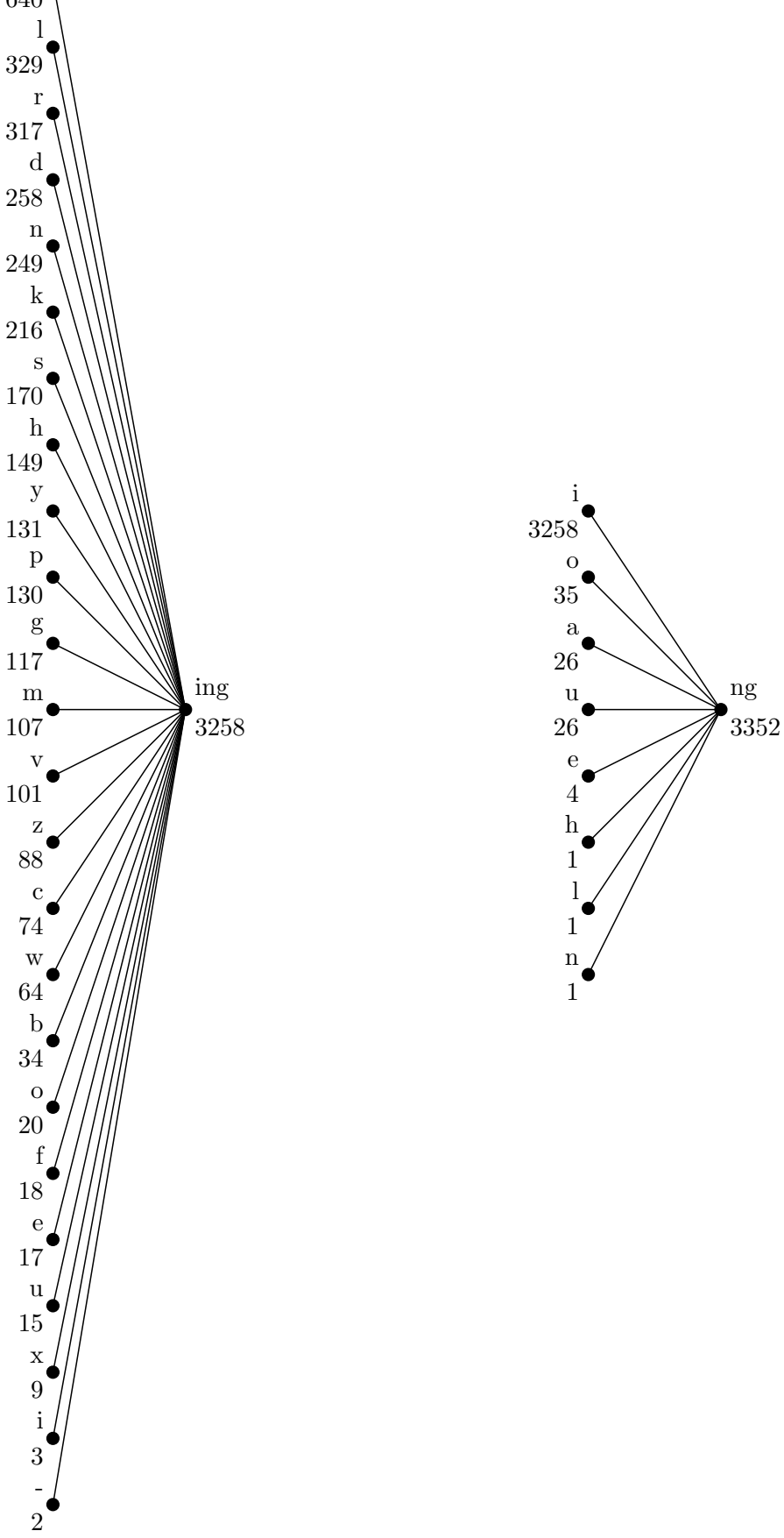

Figure 3.1: The curve frequencies giving rise to the curves C_{ing} and C_{ng} respectively.

f_W	$\overline{C}$	RA	Example	Label
high	high	high	*-ing*	True suffix
high	high	low	*-a*	Frequent random segment
high	low	high	*-ng*	Tail of true suffix
high	low	low	N/A	Second part of a digraph
low	high	high	*-oholic*	Infrequent true suffix
low	high	low	*-we*	Happenstance low RA-segment?
low	low	high	*-icz*	Tail of foreign personal name ending
low	low	low	*-ebukadnessar*	Infrequent segment

Table 3.2: The logically possible configurations of the three suffix properties, accompanied by an appropriate linguistically inspired label and an example from English.

it has some drawbacks. First, it treats all segments of the same length the same, which may be too brutal, e.g., should *-s* be penalized as much as *-a*? Second, it might be considered too vulnerable to orthography. For example if a language has an odd trigraph for some phoneme, we are clearly going to introduce an error source. Instead we propose that a segment is random iff it has similar probability in any position of the word. This avoids the "flat length"-problems but has others, which we think are less harmful. First, we might get sparse data which can either be back-off smoothed or, like here, effectively ignored (where we lack occurrence we set the RA to 1.0). Second, phonotactic or orthographic constraints may cause curiosities, e.g. English y is often spelled i when medial as in *fly* vs. *flies*.

To put it all together, we propose the characterization of suffixes in terms of the three properties as shown in table 3.2. The terms high and low are of course idealized, as they are really gradient properties.

As seen from the table, we hold that true suffixes (and only true suffixes) are those which have a high value for all three properties. Therefore, we define our final ranking score, the $Z_W : S_W \to \mathbf{Q}$:

$$Z_W(s) = \overline{C}(s) \cdot RA(s) \cdot f_W(s) \tag{3.1}$$

Thus we are deliberate saying that if you have a not-so-high relative value for one of the properties, you can compensate to some extent by having very high relative values for the other properties (relative here means relative to the corresponding values of other suffixes). It is instructive to look at what happens in a few interesting cases:

1. We have two suffixes such that one is an enlargement of the other by a random segment, e.g., *-ting* versus *-ing*, where the true suffix is the shorter one.

   In this case, we expect both to have similar high $\overline{C}$, the longer one should have higher RA and, by necessity, the shorter one should have

significantly higher frequency. Example values for *-ing* versus *-ting* are shown in table 3.3.

Here, we see that the shorter wins out and we can use that fact to weed out the longer one (cf. purging below). (One might think that in a "perfect" situation, the f_W and RA would cancel out, leaving the situation a tie. However, RA will not cancel f_W in a language which, like all language I know of, has more non-final than final "positions of segments in words", and also, ceteris paribus, we expect a higher frequency to yield a more reliable curve drop value.)

2. We have two suffixes such that one is a tail of the other, but both are true suffixes, and they just happen to share a segment, e.g., *-ly* versus *-y*.

   In this case, we succeed in keeping both if the longer wins out on a better curve-drop and random adjustment. In fact, as shown in table 3.3 this is exactly what happens with *-ly* versus *-y*.

3. We have two true suffixes which incidentally share an ending which is not a true suffix. Although easy to find in other languages, I failed to find an example of this in English without confounding factors, but we can imagine one, for example *-xz* versus *-yz*. Given the assumption that *-z* itself is not a true suffix, $f_W(z)$ should not be many times higher than $f_W(xz) + f_W(yz)$, thus its curve-drop not many percent, if at all, higher than 0.5, and of course, $RA(z) \approx 1$. On the other hand, by assumption of being true suffixes, *-xz* and *-yz* should have high curve-drop values, and outperform *-z* on RA.

   Empirically, the prediction is wrong in the case *-est* versus *-est*, as shown in 3.3 but *-ist* and *-est* can hardly be said to satisfy FFA.

4. We have two true stacked suffixes which share an ending and this ending is also true suffix, e.g., *-ations* versus *-ings*.

   As opposed to the above case, *-s* will appear in a lot of other places than after *-ing* and *-ation*, and is consequently given a higher score as shown in table 3.3.

As these considerations exemplify, the formal criterion mostly conforms to linguistic analysis, but as noted as noted in the third example, the outcomes occasionally disconcords with linguistic analysis.

A theoretical weakness with the RA-value as computed at present is when applied to languages which stack suffixes after each other. English does this to a small extent, as in *-ing* vs. *-ings*. In such cases, when calculating the non-final frequency of *-ing* one would like to count an occurrence of *-ing-* in *-ings* as a final occurrence. But this would require knowing beforehand that *-s* is a true suffix as opposed to *-ings*. Fortunately, the impact of this

s	$f_W(s)$	$\overline{C}(s)$	$RA(s)$	$Z_W(s)$
-ing	3258	0.83	19.6	53309.3
-ting	640	0.69	31.5	13929.5
-y	3931	0.63	5.8	14402.7
-ly	1532	0.76	23.4	27282.2
-t	2796	0.74	0.50	1040.6
-st	561	0.64	0.68	246.3
-ist	202	0.81	1.29	213.9
-est	213	0.88	1.82	341.4
-s	11220	0.80	2.49	22514.8
-ings	205	0.89	60.5	11034.2
-ations	215	0.86	110.9	20482.1

Table 3.3: Values for some borderline cases.

drawback, also in other languages such as Turkish, appears not to be crucial. Even if suffixes occur when they are "almost" final, they still don't occur in the initial or mid-span of the word.

As a last discussion note, it is tempting to leave out the f_W-component in the calculation of the ranking. The frequency is really only needed when deciding between suffixes which are tails of each other – it plays no crucial role in ranking between suffixes which don't share a tail. If frequencies are used only to purge out losers in tail-indexed sets of suffixes, the resulting list will also contain some non-FFA true suffixes but also too many spurious things, such as foreign personal name endings.

To sum up, the final Z_W-score in equation 3.1 is the one that purports to have the property that $Z_W(x) > Z_W(y)$ if $x \in S_W$ and $y \notin S_W$ – at least if purged (see below). We cannot give a formal proof that languages satisfying ACA and FFA should get a faultless ranking list because this is true only in a heuristic sense. To set bounds on the probability for it to hold is also depends on a lot of factors that are hard, or at least inelegant, to characterize. We hope, however, to have sketched how the ACA and FFA assumptions are used.

A summary of the algorithm described in this section is displayed in table 3.4.

The time-complexity bounding factor is the number of (final and non-final) segments, which is linear (in the size of the input) if words are bounded in length by a constant and quadratic if not.

3.4 Experimental Results

For an English bible corpus (King James 1977) we get the top 30 plus bottom 3 suffixes as shown in table 3.5.

Input: A text corpus C

Step 1. Extract the set of words W from C (thus all contextual and word-frequency information is discarded)

Step 2. Calculate $f_W(s)$, $\overline{C}(s)$ and $RA(s)$ for each $s \in S_W$

Step 3. Combine $Z_W(s) = \overline{C}(s) \cdot RA(s) \cdot f_W(s)$

Table 3.4: Summary of affix-extraction algorithm.

English has little affixation compared to e.g., Turkish which is at the opposite end of the typological scale (Dryer 2005). The corresponding results for Turkish on a bible corpus (American Bible Society 1988) is shown in table 3.6.

The results largely speak for themselves but some comments are in order. As is easily seen from the lists, some suffixes are suffixes of each other so one could *purge* the list in some way to get only the most "competitive" suffixes. One purging strategy would be to remove x from the list if there is a z such that $x = yx$ and $Z_W(z) > Z_W(x)$ (this would remove e.g., *-ting* if *-ing* is above it on the list). A more sophisticated purging method is the following, which does slightly more. First, for a word $w \in W$ define its best segmentation as: $Segment(w) = argmax_{s \lhd w} Z_W(s)$. Then purge by keeping only those suffixes which are the best parse for at least one word: $S'_W = \{s \in S_W | \exists w Segment(w) = s\}$.

Such purging kicks out the bulk of "junk" suffixes. Table 3.7 shows the numbers for English, Turkish and the virtually affixless Maori (Bauer et al. 1993). It should noted that "junk" suffixes still remain after purging – typically common stem-final characters – and that there is no simple relation between the number of suffixes left after purging and the amount of morphology of the language in question. Otherwise we would have expected the morphology-less Maori to be left with no, or 28-ish, suffixes or at least less than English.

A good sign is that the purged list and its order seems to be largely independent of corpus size (as long as the corpus is not very small) but we do get some significant differences between bible English and newspaper English.

We have chosen to illustrate using affixes but the method readily generalizes to prefixes as well and even prefixes and suffixes at the same time. As an example of this, we show top-10 purged prefix-suffix scores in the same table also for some typologically differing languages in table 3.8. Again, we use bible corpora for cross-language comparability (Swedish (Svenska Bibelsällskapet 1917) and Swahili (British and Foreign Bible Society 1953)). The scores have been normalized in each language to allow cross-language

-ed	15448.4	*-s*	3407.3
-eth	12797.1	*-ions*	2684.5
-ted	11899.4	*-est*	2452.6
-iah	11587.5	*-sed*	2313.7
-ly	10571.2	*-y*	2239.2
-ings	8038.9	*-leth*	2166.3
-ing	7292.8	*-nts*	2122.6
-ity	6917.6	*-ied*	1941.7
-edst	6844.7	*-ened*	1834.9
-ites	5370.2	*-ers*	1819.5
-seth	5081.6	*-ered*	1796.7
-ned	4826.7	*-ded*	1582.2
-s'	4305.2	*-neth*	1540.0
-nded	3833.8	...	...
-ts	3783.1	*-ig*	0.0
-ah	3766.9	*-io*	0.0
-ness	3679.3	*-ti*	0.0

Table 3.5: Top 30 and bottom 3 extracted suffixes for an English bible corpus. The high placement of English *-eth* and *-iah* are due to the fact that the bible version used has drinketh, sitteth etc and a lot of personal names in *-iah*.

comparison – which, judging from the table, seems meaningful. Swahili is an exclusively prefixing language but verbs tend to end in *-a* (whose status as a morpheme is the linguistic sense can be doubted), whereas Swedish is suffixing, although some prefixes are or were productive in word-formation.

A full discussion of further aspects such as a more informed segmentation of words, peeling of multiple suffix layers and purging of unwanted affixes requires, is beyond the scope of this paper.

3.5 Related Work

For reasons of space we cannot cite and comment every relevant paper even in the narrow view of highly unsupervised extraction of affixes from raw corpus data, but we will cite enough to cover each line of research. The vast fields of word segmentation for speech recognition or for languages which do not mark word boundaries will not be covered. In our view, segmentation into lexical units is a different problem than that of affix extraction since the frequencies of lexical items are different, i.e., occur much more sparsely. Results from this area which have been carried over or overlap with affix detection will however be taken into account. A lot of the papers cited have a wider scope and are still useful even though they are critisized here for

-larına	71645.4	*-adılar*	16587.9
-larından	47941.9	*-lerinden*	15201.1
-lerinin	43917.3	*-nden*	14082.2
-lerden	36294.0	*-sinin*	13493.9
-inden	35258.2	*-nin*	12340.9
-iyorlardı	28716.2	*-yorsunuz*	12135.0
-arak	27774.1	*-larla*	12069.7
-iyorsunuz	25403.1	*-en*	11513.5
-inin	25045.5	*-ten*	11424.0
-dılar	20718.7	*-siniz*	11043.0
-lere	20718.2	*-madılar*	10958.9
-ip	20431.2	*-lardan*	10428.1
-dan	19468.4	*-siniz*	10391.1
-ndan	18556.3	*-...*	...
-ından	18226.3	*-ist*	0.0
-yorlardı	18097.1	*-iy*	0.0
-acaksınız	16751.1	*-yo*	0.0

Table 3.6: Top 30 and bottom 3 extracted suffixes for Turkish. Most of these are really compounds of two suffixes, showing that some adaptation to multi-layer suffixing languges is appropriate.

Language	Corpus	Tokens	$\|W\|$	$\|S_W\|$	$\|S'_W\|$
Maori	(The British & Foreign Bible Society 1996)	1101665	8354	23007	78
English	(King James 1977)	917634	12999	39845	63
Turkish	(American Bible Society 1988)	574592	56881	175937	122

Table 3.7: Figures for different languages on the effects on the size of the suffix list after purging.

having a non-optimal affix detection component.

Many authors trace their approaches back to two early papers by Zellig Harris (Harris 1955: 1970) which count *letter successor varieties*. The basic procedure is to ask how many different phonemes occur (in various utterances e.g., a corpus) after the first n phonemes of some test utterance and predict that segmentation(s) occur where the number of successors reaches a peak. For example, if we have *play*, *played*, *playing*, *player*, *players*, *playground* and we wish to test where to segment *plays*, the successor count for the prefix *pla* would be 1 because only y occurs after whereas the number of successors of *play* peak at three (i.e., $\{e, i, g\}$). Although the heuristic has had some success it was shown (in various interpretations) as early as Hafer and Weiss 1974 that it is not really sound – even for English. A slightly better method is to compile a set of words into a *trie* and predict boundaries at nodes with high activity (e.g. Johnson and Martin 2003; Schone

Swedish		English		Swahili	
för-	0.097	*-ed*	0.132	*-a*	0.100
-en	0.086	*-eth*	0.109	*wa-*	0.095
-na	0.036	*-iah*	0.099	*ali-*	0.065
-ade	0.035	*-ly*	0.090	*nita-*	0.059
-a	0.034	*-ings*	0.068	*aka-*	0.049
-ar	0.033	*-ing*	0.062	*ni-*	0.046
-er	0.033	*-ity*	0.059	*ku-*	0.044
-as	0.032	*-edst*	0.058	*ata-*	0.042
-s	0.031	*-ites*	0.046	*ha-*	0.032
-de	0.031	*-s'*	0.036	*a-*	0.031
...	...	...	...	...	...

Table 3.8: Comparative figures for prefix vs. suffix detection.

and Jurafsky 2001; Kazakov and Manandhar 2001 and earlier papers by the same authors), but this not sound either as non-morphemic short common character sequences also show significant branching.

The algorithm in this paper is differs significantly from the Harris-inspired varieties. First, we do not record the number of phonemes/character of a given prefix/suffix but their frequency distribution. In the example above, that would be the distribution { *e:3*, *i:1*, *g:1* } rather than a uniform three-member set. Secondly, segmentation of a given word is not the immediate objective and what amounts to identification of the end of a lexical (thus generally low-frequency) item is not within the direct reach of the model. Thirdly, and most importantly, the algorithm in this paper looks at the *relative drop* of the frequency curve not at peaks in absolute frequency.

A different approach, sometimes used in complement of other sources of information, is to select *aligned pairs* (or sets) of strings that share a long character sequence (work includes Jacquemin (1997); Yarowsky and Wicentowski (2000); Baroni et al. (2002); Clark (2001a)). A notable advantage is that one is not restricted to concatenative morphology.

Many publications (Ćavar et al. 2004; Brent et al. 1995; Goldsmith et al. 2001; Déjean 1998a; Snover et al. 2002; Argamon et al. 2004; Goldsmith 2001; Creutz and Lagus 2005b; Neuvel and Fulop 2002; Baroni 2003; Gaussier 1999; Sharma et al. 2002; Wicentowski 2002; Oliver 2004), and various other works by the same authors, describe strategies that use frequencies, probabilities, and optimization criteria, often Minimum Description Length (MDL), in various combinations. So far, all these are unsatisfactory on two main accounts; on the theoretical side, they still owe an explanation of why compression or MDL should give birth to segmentations coinciding with morphemes as linguistically defined. On the experimental side, thresholds, supervised/developed parameters and selective input still cloud the success

of reported results, which, in any case, aren't wide enough to sustain some too rash language independence claims.

To be more specific, some MDL approaches aim to minimize the description of the set of words in the input corpus, some to describe all tokens in the corpus, but, none aims to minimize, what one would otherwise expect, the set of possible words in the language. More importantly, none of the reviewed works allow any variation in the description language ("model") during the minimization search. Therefore they should be more properly labelled "weighting schemes" and it's an open question whether their yields correspond to linguistic analysis. Given an input corpus and a traditional linguistic analysis, it is trivial to show that it is possible to decrease description length (according to the given schemes) by stepping away from linguistic analysis. Moreover, various forms of codebook compression, such as Lempel-Ziv compression, yield shorter descriptions but without any known linguistic relevance at all. What is clear, however, apart from whether it is theoretically motivated, is that MDL approaches are *useful*.

A systematic test of segmentation algorithms over many different types of languages has yet to be published. For three reasons, it will not be undertaken here either. First, as e.g., already Manning (Manning 1998) notes for sandhi phenomena, it is far from clear what the gold standard should be (even though we may agree or agree to disagree on some familiar European languages). Secondly, segmentation algorithms may have different purposes and it might not make good sense to study segmentation in isolation from induction of paradigms. Lastly, and most importantly, all of the reviewed techniques (Wicentowski 2004: 2002; Snover et al. 2002; Baroni et al. 2002; Andreev 1965; Ćavar et al. 2004; Snover and Brent 2003: 2001; Snover 2002; Schone and Jurafsky 2001; Jacquemin 1997; Goldsmith and Hu 2004; Sharma et al. 2002; Clark 2001a; Kazakov and Manandhar 1998; Déjean 1998a; Oliver 2004; Creutz and Lagus 2002: 2004; Hirsimäki et al. 2003; Creutz and Lagus 2005b; Argamon et al. 2004; Gaussier 1999; Lehmann 1973; Langer 1991; Flenner 1995; Klenk and Langer 1989; Goldsmith 2001: 2000; Hu et al. 2005b:a; Brent et al. 1995), as they are described, have threshold-parameters of some sort, explicitly claim **not** to work well for an open set of languages, or require noise-free all-form input (Albright 2002; Manning 1998; Borin 1991). Therefore it is not possible to even design a fair test.

In any event, we wish to appeal to the merits of developing a theory in parallel with experimentation – as opposed to only ad hoc result chasing. If we have a theory and we don't get the results we want, we may scrutinize the assumptions behind the theory in order to modify or reject it (understanding why we did so). Without a theory there's no telling what to do or how to interpret intermediate numbers in a long series of calculations.

3.6 Conclusion

We have presented a new theory of affixation and a parameter-less efficient algorithm for collecting affixes from raw corpus data of an arbitrary language. Depending on one's purposes with it, a cut-off point for the collected list is still missing, or at least, we do not consider that matter here. The results are very promising and competitive but at present we lack formal evaluation in this respect. Future directions also include a more specialized look into the relation between affix-segmentation and paradigmatic variation and further exploits into layered morphology.

3.7 Acknowledgements

The author has benefited much from discussions with Bengt Nordström. The author is also grateful to Bob Carpenter for pointing out a grave technical error in an earlier version of this paper.

Chapter 4

Paper III: Poor Man's Stemming: Unsupervised Recognition of Same-Stem Words

Harald Hammarström
Department of Computing Science
Chalmers University of Technology
harald2@cs.chalmers.se

Abstract

We present a new fully unsupervised human-intervention-free algorithm for stemming for an open class of languages. Since it does not rely on existing large data collections or other linguistic resources than raw text it is especially attractive for low-density languages. The stemming problem is formulated as a decision whether two given words are variants of the same stem and requires that, if so, there is a concatenative relation between the two. The underlying theory makes no assumptions on whether the language uses a lot of morphology or not, whether it is prefixing or suffixing, or whether affixes are long or short. It does however make the assumption that 1. salient affixes have to be frequent, 2. words essentially are variable length sequences of random characters, and furthermore 3. that a heuristic on what constitutes a systematic affix alteration is valid. Tested on four typologically distant languages, the stemmer shows very promising results in an evaluation against a human made gold standard.

4.1 Introduction

The problem at hand can be described as follows:

Input : An unlabeled corpus of an arbitrary natural language and two arbitrary words w_1, w_2 from that language

Output : A YES/NO answer as to whether w_1 and w_2 are morphological variants of one and the same stem (according to traditional linguistic analysis).

Restrictions : We consider only concatenative morphology and assume that the corpus comes already segmented on the word level.

The relevance of the problem is that of stemming as applied in Information Retrieval (IR). The issues of stemming in IR has been discussed at length elsewhere and need not be repeated here. It suffices to say that, though not uncontroversial, stemming continues to be a feature of modern IR systems for languages like English (e.g., Google[1]), and is likely to be of crucial importance for languages which make more use of morphology (cf. Pirkola 2001).

The reasons for attacking the problem in an unsupervised manner include advantages in elegance, economy of time and money (no annotated resources required), and the fact that the same technology may be used on new languages. The latter two reasons are especially important in the context of resource-scarce languages.

Our proposed unsupervised same-stem decision algorithm proceeds in two phases. In the first phase, a ranked list of salient affixes are extracted from an unlabeled text corpus of a language. In the second phase, an input word pair is aligned to shortlist affixes that could potentially be added to a common stem to alternate between the two. Crucially, this shortlist of affix alternations is analyzed to check whether they form a *systematic* alternation in the language as a whole (i.e., not just in the pair at hand). This analysis depends strongly on the ranked affix list from the first phase.

An outline of the paper is as follows: we start with some notation and basic definitions, with which we describe the theory that is intended to model the assumed behaviour of affixation in natural languages. Then we describe in detail and with examples the thinking behind the affix extraction phase, which actually requires only a few lines to define mathematically. Following that, we present our ideas on how to distinguish a systematic morphological alternation from a spurious one. This part is the more experimental one but at least it requires no guiding, tuning or annotation whatsoever. The algorithm is evaluated against a human gold standard on four languages chosen to span the full width of morphological typology. Finally, we briefly

[1] According to http://www.google.com/help/basics.html accessed 20 March 2006.

discuss related work, draw some tentative conclusions and hint at future directions.

4.2 Affix Extraction

We have chosen to illustrate using suffixes but the method readily generalizes to prefixes as well (and even prefixes and suffixes at the same time).

4.2.1 A Theory of Affixation

Notation and definitions:

- $w, s, b, x, y, \ldots \in \Sigma^*$: lowercase-letter variables range over strings of some alphabet Σ and are variously called words, segments, strings, etc.
- $s \triangleleft w$: s is a terminal segment of the word w i.e., there exists a (possibly empty) string x such that $w = xs$
- $W, S, \ldots \subseteq \Sigma^*$: capital-letter variables range over sets of words/strings/segments
- $f_W(s) = |\{w \in W | s \triangleleft w\}|$: the number of words in W with terminal segment s
- $S_W = \{s | s \triangleleft w \in W\}$: all terminal segments of the words in W
- $|\cdot|$: is overloaded to denote both the length of a string and the cardinality of a set

Assume we have two sets of random strings over some alphabet Σ:

- Bases $B = \{b_1, b_2, \ldots, b_m\}$
- Suffixes $S = \{s_1, s_2, \ldots, s_n\}$

Such that:

Arbitrary Character Assumption (ACA) Each character $c \in \Sigma$ should be equally likely in any word-position for any member of B or S.

Note that B and S need not be of the same cardinality and that any string, including the empty string, could end up belonging to both B and S. They need neither to be sampled from the same distribution; pace the requirement, the distributions from which B and S are drawn may differ in how much probability mass is given to strings of different lengths. For instance, it would not be violation if B were drawn from a distribution favouring strings of length, say, 42 and S from a distribution with a strong bias for short strings.

Next, build a set of affixed words $W \subseteq \{bs | b \in B, s \in S\}$, that is, a large set whose members are concatenations of the form bs for $b \in B, s \in S$, such that:

Frequent Flyer Assumption (FFA) : The members of S are frequent. Formally: Given any $s \in S$: $f_W(s) >> f_W(x)$ for all x such that 1. $|x| = |s|$; and 2. not $x \triangleleft s'$ for all $s' \in S$).

In other words, if we call $s \in S$ a *true suffix* and we call x an *arbitrary segment* if it neither a true suffix nor the terminal segment of a true suffix, then any true suffix should have much higher frequency than an arbitrary segment of the same length.

One may legimately ask to what extent words of real natural languages fit the construction model of W, with the strong ACA and FFA assumptions, outlined above. For instance, even though natural languages often aren't written phonemically, it is not hard to come up with languages that have phonotactic constraints on what may appear at the beginning or end of a word, e.g., Spanish **st-* may not begin a word and yields *est-* instead. Another violation of ACA is that (presumably all (Ladefoged 2005)) languages disallow or disprefer a consonant vs. a vowel conditioned by the vowel/consonant status of its predecessor. However, if a certain element occurs with *less* frequency than random (the best example would be click consonants which, in some languages e.g., Eastern !Xõó (Traill 1994), occur only initially), this is not a problem to the theory.

As for FFA, we may have breaches such as Biblical Aramaic (Rosenthal 1995) where an old -$\bar{a}$ element appears virtually everywhere on nouns, making it very frequent, but no longer has any synchronic meaning. Also, one can doubt the requirement that an affix should need to be frequent; for instance, the Classical Greek inflectional (lacking synchronic internal segmentation) alternative medial 3p. pl. aorist imperative ending -$\sigma\theta\omega\nu$ (Blomqvist and Jastrup 1998), is not common at all.

Eaxctly how realistic the assumptions are is an empirical question, whose answer must be judged by experiments on the relevant languages.

4.2.2 An Algorithm for Affix Extraction

The key question is, if words in natural languages are constructed as W explained above, can we recover the segmentation? That is, can we find B and S, given only W? The answer is yes, we can partially decide this. To be more specific, we can compute a score Z such that $Z(x) > Z(y)$ if $x \in S_W$ and $y \notin S_W$. In general, the converse need not hold, i.e., if both $x, y \in S_W$, or both $x, y \notin S_W$, then it may still be that $Z(x) > Z(y)$. This is equivalent to constructing a ranked list of all possible segments, where the true members of S_W appear at the top, and somewhere down the list the junk, i.e., non-members of S_W, start appearing and fill up the rest of

the list. Thus, it is not said *where* on the list the true-affixes/junk border begins, just that there is a consistent such border.

Now, how should this list be computed? Given the FFA, it's tempting to look at frequencies alone, i.e., just go through all words and make a list of all segments, ranking them by frequency? This won't do it because 1. it doesn't compensate between segments of different length; naturally, short segments will be more frequent than long ones, solely by virtue of their shortness 2. it overcounts ill-segmented true affixes, e.g., *-ng* will invariably get a higher (or equal) count than *-ing*. What we will do is a modification of this strategy, because 1. can easily be amended by subtracting estimated prior frequencies (under ACA) and there is a clever way of tackling 2. Note that, to amend 2., when going through w and each $s \triangleleft w$, it would be nice if we could count s only when it is well-segmented in w. We are given only W so this information is not available to us, but, the FFA assumption let's us make a local guess of it.

We shall illustrate the idea with an example of an evolving frequency curve of a word "playing" and its segmentations "playing", "aying", "ying", "ing", "ng", "g" (W being the set of words from an English bible corpus (King James 1977)). Figure 4.1 shows a frequency curve $f_W(s)$ and its expected frequency curve $e_W(s)$. The expected frequency of a suffix s doesn't depend on the actual characters of s and is defined as:

$$e_W(s) = |W| \cdot \frac{1}{r^{|s|}}$$

Where r is the size of the alphabet under the assumption that its characters are uniformly distributed. We don't simply use 26 in the case of lowercase English since not all characters are equally frequent. Instead we estimate the size of a would-be uniform distribution from the entropy of the distribution of the characters in W. This gives $r \approx 18.98$ for English and other languages with a similar writing practice.

Next, define the adjusted frequency as the difference between the observed frequency and the expected frequency:

$$f'_W(s) = f_W(s) - e_W(s)$$

It is the slope of this curve that predicts the presence of a good split. Figure 4.2 shows the appearance of this curve again exemplified by "playing".

After these examples, we are ready to define the segmentation score of a suffix relative to a word $Z : S_W \times W \to \mathbf{Q}$:

$$Z_W(s, w) = \begin{cases} 0 & \text{if not } s \triangleleft w \\ \frac{f'(s_i) - f'(s_{i-1})}{|f'(s_{i-1})|} & \text{if } s = s_i(w) \text{ some } i \end{cases}$$

Table 4.1 shows the evolution of exact values from the running example.

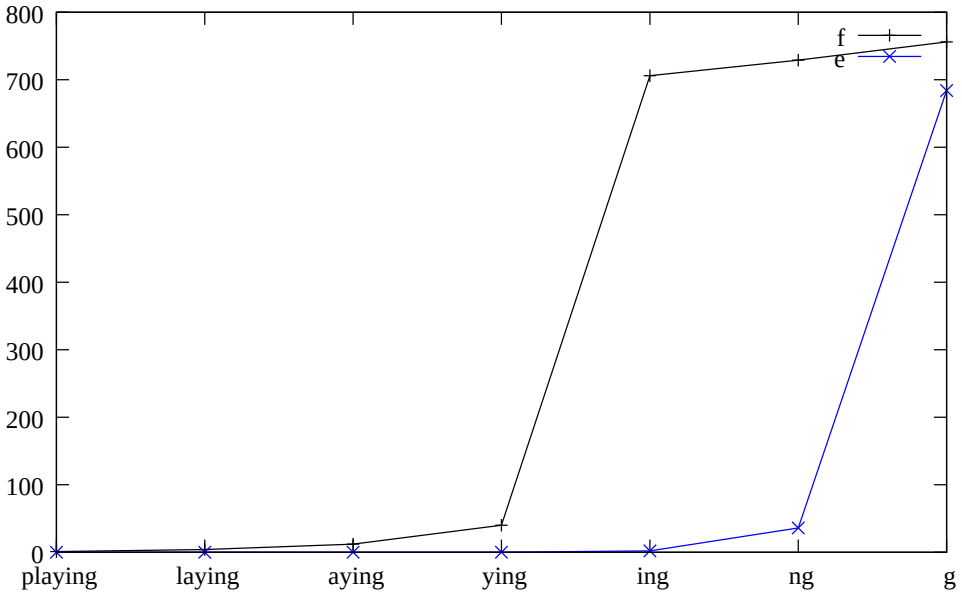

Figure 4.1: The observed $f_W(s)$ and expected $e_W(s)$ frequency for $s \triangleleft w = \textit{playing}$.

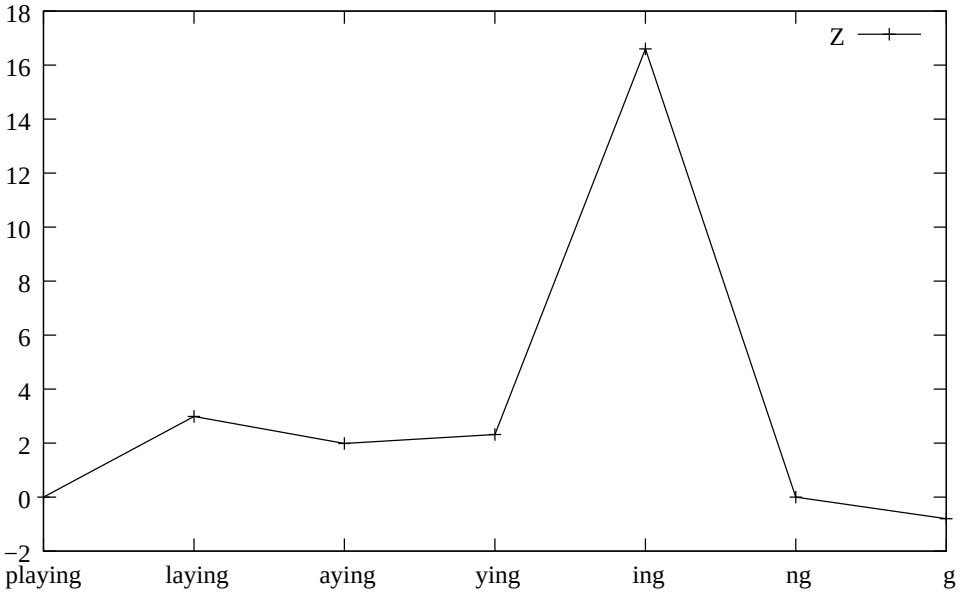

Figure 4.2: The slope of the $f'_W(s)$ curve for $s \triangleleft w = \textit{playing}$.

s	*playing*	*laying*	*aying*	*ying*	*ing*	*ng*	*g*
$f(s)$	1	4	12	40	706	729	756
$e_W(s)$	0.00	0.00	0.00	0.10	1.90	36.0	684
$f(s) - e_W(s)$	0.99	3.99	11.9	39.8	704	692	71.0
$Z(s,$"playing"$)$	**0.00**	**2.99**	**1.99**	**2.32**	**16.6**	**-0.0**	**-0.8**

Table 4.1: Exact values of frequency curves and scores from the running "playing" example.

Input: A text corpus C

Step 1. Extract the set of words W from C (thus all contextual and word-frequency information is discarded)

Step 2. Calculate $Z_W(s, w)$ for each $w \in W$ and $s \triangleleft w$

Step 3. Accumulate $Z_W(s) = \sum_{w \in W} Z(s, w)$

Table 4.2: Summary of affix-extraction algorithm.

To move from a Z-score for a segment that is relative to a word we simply sum over all words to get the final score $Z : S_W \to \mathbf{Q}$:

$$Z_W(s) = \sum_{w \in W} Z(s, w) \tag{4.1}$$

To be extra clear, the FFA assumption is "exploited" in two ways. On the one hand, frequent affixes get many oppurtunities to get a score (which could, however, be negative) in the final sum over $w \in W$. On the other hand, the frequency is what make up the appearance of the slope that predicts the segmentation point.

The final Z-score in equation 4.1 is the one that purports to have the property that $Z(x) > Z(y)$ if $x \in S_W$ and $y \notin S_W$ – at least if purged (see below). A summary of the algorithm described in this section is displayed in table 4.2.

The time-complexity bounding factor is the number of suffixes, i.e., the cardinality of S_W, which is linear in the size of the input.

4.2.3 Affix Extraction Sample Results

On the affix extraction part as such, we will only give some impressionistic results rather than a full-scale evaluation. The reason for this is that, although undoubtedly the list has some valid meaning, it is at present unclear to the author what a gold standard should be in every detail in every language. Furthermore, different applications, such as the final objective in

this paper, may not require that a context-less choice between two related affixes, e.g., *-ation* and *-tion*, be asserted.

For a regular English 1 million token newspaper corpus we get the top 30 plus bottom 3 suffixes as shown in table 4.3.

1028682.0	*ing*	111264.0	*ling*
594208.0	*ed*	111132.0	*ent*
371145.0	*s*	109725.0	*ating*
337464.0	*'s*	109125.0	*ate*
326250.0	*ation*	108228.0	*an*
289536.0	*es*	97020.0	*ies*
238853.5	*e*	94560.0	*ts*
222256.0	*er*	81648.0	*ically*
191889.0	*ers*	81504.0	*ment*
172800.0	*ting*	78669.0	*led*
168288.0	*ly*	77900.0	*ering*
159408.0	*ations*	74976.0	*er's*
143775.0	*ted*	73988.0	*y*
130960.0	*able*	...	...
116352.0	*ated*	-26137.9	*l*
113364.0	*al*	-38620.6	*m*
113280.0	*ness*	-78757.3	*a*

Table 4.3: Top 30 and bottom 3 extracted suffixes for English. 47178 unique words yielded a total of 154407 ranked suffixes.

English has little affixation compared to e.g., Turkish which is at the opposite end of the typological scale (Dryer 2005). The corresponding results for Turkish on a bible corpus (American Bible Society 1988) is shown in table 4.4.

The results largely speak for themselves but some comments are in order. A good sign is that the purged list and its order seems to be largely independent of corpus size (as long as the corpus is not extremely small) but we do get some significant differences between bible English and newspaper English. As is easily seen from the lists, some suffixes are suffixes of each other so one could *purge* the list in some way to get only the most "competitive" suffixes. A full discussion of further aspects such as a more informed segmentation of words, peeling of multiple suffix layers and purging of unwanted affixes requires, is beyond the scope of this paper.

1288402.4	*i*	33756.55	*ler*
151056.9	*er*	29816.53	*da*
142552.6	*in*	29404.49	*di*
141603.3	*im*	28337.89	*le*
134403.2	*en*	26580.41	*dan*
130794.5	*e*	26373.54	*r*
127352.0	*an*	24183.99	*ti*
113482.6	*a*	22527.26	*un*
82581.95	*ya*	21388.71	*iniz*
78447.74	*ar*	20993.87	*sin*
76353.77	*ak*	20117.60	*ik*
68730.00	*n*	18612.14	*li*
64761.37	*ir*	18316.45	*ek*
53021.67	*la*	...	...
47218.78	*ini*	-38091.8	*t*
44858.18	*lar*	-240917.5	*l*
37229.14	*iz*	-284460.1	*s*

Table 4.4: Top 30 and bottom 3 extracted suffixes for Turkish. 56881 unique words yielded a total of 175937 ranked suffixes.

4.3 Affix Alternation Analysis

Having a list of salient affixes is not sufficient to parse a given word into stem and affix(es). For example, *sing* happens to end in the most salient suffix yet it is not composed of *s* and *ing* because crucially, there is no **s*, **sed* etc. Thus to parse a given word we have to look at additional evidence beyond the word itself, such as the existence of other inflections of potentially the same stem as the given word, or further, look at inflections of other stems which potentially share an affix with the given word. This line of though will be pursued below.

The problem at hand, namely, to decide if two given words w_1, w_2 share a common stem (in the linguistic sense) is easier than parsing one word since we have evidence from two words. Essentially, there are four interesting kinds of situations the same-stem-decider must face:

1. w_1 and w_2 do share the same stem and have a salient affix each, e.g., *played* vs. *playing*.

2. w_1 and w_2 do share the same stem but one of them has the "zero" affix, e.g., *play* vs. *playing*.

3. w_1 and w_2 don't share the same stem (linguistically) but do share some initial segment, e.g., *playing* vs. *plough*.

4. w_1 and w_2 don't share the same stem (linguistically) and don't share any initial segment, e.g., *playing* vs. *song*.

Number 4 is trivial to decide in the negative. Number 1 is also easy to affirm using a list of salient affixes, whereas the special case of number 2 requires some care. The real difficulty lies in predicting a negative answer for case number 3 (while, of course, at the same time predicting a positive for cases 1 and 2). We will go for an extended discussion of this matter below.

Consider two words $w_1 = xs_1$ and $w_2 = xs_2$ that share some non-empty initial segment x. Except for chance resemblances, which by definition are rare, we would like to say that w_1 and w_2 belong to the same stem iff:

1. s_1 and s_2 are well-segmented salient suffixes in the language, i.e., *-w* and *-lt* for *saw* and *salt* are **not**; and

2. s_2 and s_2 must systematically contrast in the language, that is, there must be a large set of stems which can take both s_1 and s_2. For example, the word pair *sting* and *station* align to *-ing* and *-ation* which are both salient suffixes but they do **not** systematically contrast.

The key difficulty is to decide, in an unsupervised manner, when something is systematic and when it isn't. In order to tackle this, we will propose a heuristic for measuring how much two suffixes contrast. This will give a score between 0 and 1 where it is not clear at which value "systematic" begins. We could say that, at this point, the user has to supply a threshold value. However, instead, we devise another heuristic that obviates the need for a threshold at all. The resulting system thus supplies a YES/NO answer to the same-stem decision problem without any human interaction.

4.3.1 Formalizing Same Stem Co-Occurrence

From the word distributions characteristic of natural language corpora, it is surprisingly difficult to come up with a measure of how much a set of suffixes show up on the "same stems" that is not such that it favours the inclusion of any simply frequent, rather than truly contrasting, terminal segment. For example, the author has not had much success with standard vector similarity measures. Instead, we propose the following usage of co-occurrence statistics. The measure presented is valid for an arbitrary set of suffixes (called P for "paradigm") even though the relevance in this paper is for the case where $|P| = 2$.

First, for each suffix x, define its quotient function $H_x(y) : S_W \rightarrow [0, 1]$ as:

$$\frac{|s|sx \in w \wedge sy \in w|}{|s|sx \in w|}$$

The formula is conveying the following: We are given a suffix x, and we want to construct a quotient function which is a function from any other

y	$H_{ing}(y)$	y	$H_{ed}(y)$
ing	1.00	*ed*	1.00
ed	0.59	*ing*	0.42
$''$	0.41	$''$	0.33
s	0.25	*e*	0.21
e	0.24	*s*	0.20
es	0.19	*es*	0.17
er	0.12	*er*	0.08
ers	0.10	*ion*	0.07
ion	0.07	*ers*	0.05
y	0.05	*y*	0.04
ings	0.05	*ions*	0.03
ions	0.03	*ation*	0.03
in	0.03	*able*	0.02
ation	0.03	*ings*	0.02
$'s$	0.03	$'s$	0.02
ingly	0.03	*or*	0.02
or	0.02	*in*	0.01
able	0.02	*ly*	0.01
ive	0.02	*ive*	0.01
ors	0.02	*ingly*	0.01
ations	0.01	*al*	0.01
$er's$	0.01	*ment*	0.01
ment	0.01	*ors*	0.01
ly	0.01	*ations*	0.01
...	...	...	...

Table 4.5: Sample quotient functions/lists for *ing* and *ed*.

suffix to a score between 0 and 1. The score is calculated as: look at all the stems of x, other suffixes y will undoubtedly also occur on some of these stems. For each other suffix y, find the proportion of x:s stems on which y also appears. This proportion will be the quotient associated with y. Two examples of quotient function (sorted on highest value) are given in table 4.5.

Now, given a set of affixes P, construct a rank by summing the quotient functions of the members of P:

$$V_P(y) = \sum_{x \neq y \in P} H_x(y)$$

The $x \neq y$ is just there so that the y:s that are also in P don't get an "extra" 1.0, since $H_x(x) = 1.0$ regardless of the data. The rank of y is the number of suffixes s with $V_P(s) < V_P(y)$ – in other words – the place of y on a list

of suffixes sorted on on highest V_P.

As an example, take W from the Swedish PAROLE-Corpus (Borin 1997). We can compare in table 4.6 the very common paradigm $\{a, an, as, ans, or, orna, ors, ornas\}$ with the nonsense paradigm $\{ungen, ig, ar, ts, s, de, ende, er\}$ consisting only of individually frequent suffixes.

y	$V_{P_1}(y)$	y	$V_{P_2}(y)$
a	3.93	”	3.32
an	2.82	t	1.48
or	2.71	a	1.19
”	1.91	r	1.18
orna	1.76	**s**	1.15
ar	1.13	en	1.14
as	1.06	iga	0.86
ade	1.05	d	0.80
ans	0.94	igt	0.73
at	0.89	as	0.66
en	0.82	**de**	0.59
s	0.76	des	0.57
t	0.73	ade	0.55
e	0.71	ung	0.49
er	0.66	**er**	0.49
ad	0.61	at	0.48
ande	0.52	n	0.46
ades	0.47	**ar**	0.45
ats	0.40	an	0.44
i	0.36	e	0.42
...	...	...	...
ors	0.35		
...	...		
ornas	0.27		
...	...		

Table 4.6: Example ranks for $P_1 = \{a, an, as, ans, or, orna, ors, ornas\}$ and $P_2 = \{ungen, ig, ar, ts, s, de, ende, er\}$. ” means the empty suffix.

In table 4.6, the ranks of the member of P_1 to the left are [0, 1, 2, 4, 6, 8, 22, 31], and for P_2 to the right the ranks are [115044, 127, 17, 28, 4, 10, 100236, 14].

Now, if we can generalize from these cases it seems that we can rank different hypotheses of paradigms (of the same size) by looking at their quotient ranks. If the members of P “turn up high in” the quotient rank then the members of P tend to turn up on the same stems. There are several issues in formalizing the notion of “turn up high in”. The ranks in the ranked

P	$VI(P)$	P	$VI(P)$
('ation',)	0.00	('xt')	0.00
('ated', 'ation')	0.14	('xt', 'n')	0.04
('ate', 'ated', 'ation')	0.40	('xt', 'n', 'ns')	0.12
('ate', 'ated', 'ating', 'ation')	0.75	('n', 'ns')	0.55
('ate', 'ated', 'ating', 'ation', 'ations')	1.00	...	...

Table 4.7: Example iterations of $G^*('ation')$ and $G^*('xt')$.

list alone? Also incorporate the scores? Average rank or total sum of ranks? For now we will just do a simple sum of ranks in the ranked list, divide by the optimum sum (which depends on $|P|$ and is $0 + \ldots + |P| - 1$), and take the inverse. This gives a score between 0 and 1 where a high score means the members of P tend to appear on the same stems:

$$VI(P) = \frac{|P|(|P|-1)}{2\sum_{x \in P} rank(x, V_P)}$$

According to the desiderata 1 and 2 in section 4.3 (p. 41) we finally define an affix-systematicity likelihood score as:

$$A(P) = VI(P) \sum_{s \in P} Z_W(s) \tag{4.2}$$

As a convention we set $Z_W('') = 0$.

4.3.2 Escaping Thresholds

The VI-score from the last section may be used for a greedy hill-climbing search through the affix set space. For example, we may start with an affix, a one member set, and see whether we can improve the affix score by including another member, and perhaps another after that until we can't improve the score anymore. In this process, we may also entertain the possibility of kicking some member out if that improves the score – as long as there is no backtracking the search remains polynomial. Formally, define the growing function of a set P of affixes as:

$$G(P) = argmax_{p \in \{P\} \cup \{P \text{ xor } s | s \in S^W\}} VI(p) \tag{4.3}$$

$$G^*(P) = \begin{cases} P & \text{if } G(P) = P \\ G^*(G(P)) & \text{if } G(P) \neq P \end{cases} \tag{4.4}$$

Two growth-examples are shown in table 4.7, one which attains a perfect 1.0 score and one in which the original member is expelled in a later iteration.

Now, how does this help us work around a threshold for deciding how systematically a pair of suffixes have to co-occur to conflate their stems? Recall the writing convention $w_1 = xs_1$ and $w_2 = xs_2$. Instead of having a threshold we may conjecture that:

Input: A text corpus C and two words w_1, w_2

Step 1. Calculate Z_W as in table 4.2

Step 2. Form the set of candidate alignment pairs as:

$$C(w_1, w_2) = \{(s1, s2) | xs_1 = w_1 \text{ and } xs_2 = w_2\} \tag{4.5}$$

Step 3. If $C(w_1, w_2)$ is empty then answer NO, otherwise pick the best candidate pair as:

$$argmax_{(s1,s2)\in C(w_1,w_2)} A(\{s_1, s_2\}) \tag{4.6}$$

Step 4. For the winning pair, answer YES/NO accordingly as $s_1 \in G^*(s_2)$ and $s_2 \in G^*(s_1)$

Table 4.8: Summary of same-stem decision algorithm

w_1, w_2 have the same stem iff $s_1 \in G^*(s_2)$ and $s_2 \in G^*(s_1)$

For example, this predicts that *sting* and *station* are not the same stem because neither $G^*(ing) = \{'', e, ed, er, es, ing, s\}$ contains 'ation' nor does $G^*(ation) = \{ate, ated, ating, ation, ations\}$ contain 'ing'. From our experience this test is quite powerful. However, there are of course cases where it predicts wrongly, due to the greedy nature of the G^*-calculation, e.g., $G^*(ing)$ does not contain 'ers'. Moreover, if one of the affixes is the empty affix, we need a special fix (see below).

4.3.3 Same-stem Decision Algorithm

We can now put all pieces together to define the full algorithm as shown in table 4.8.

If one of s_1, s_2 is the empty string then step 3 and 4 should be restated as follows (using s to denote the non-empty one of the two). The maximization value in step 3 should be modified to: $\frac{Z_W(s)}{1+rank('',H_s)}$. Step 4 should be modified to: answer YES/NO accordingly as $'' \in G^*(s)$.

The bad news is that the computation of the G^*:s tends to be slow due to the summing and sorting of typically very long (50 000-ish items) lists. On my standard PC with a Python implementation it typically takes 30 seconds to decide whether two words share the same stem.

Language	Language Type	Corpus	Scope
Maori	Isolating	(The British & Foreign Bible Society 1996)	NT & OT
English	Mildly Suffixing	(King James 1977)	NT & OT
Swedish	Suffixing	(Svenska Bibelsällskapet 1917)	NT & OT
Kuku Yalanji	Strongly Suffixing	(Summer Institute of Linguistics 1985)	NT & OT Parts

Table 4.9: Summary of Bible corpora used.

4.4 Evaluation

Several authors, e.g., Goldsmith et al. (2001); Melucci and Orio (2003), have evaluated their stemming algorithms on Information Retrieval performance. While IR is the undoubtedly the major application area we feel that evaluating on retrieval performance does not answer all relevant questions of stemming performance. For instance, a stemmer may make conflations and miss conflations that simply did not affect the test queries. In fact, one may get different best stemmers depending on the test collection. There is also difference as to whether the whole document collection, an abstract of each document or just the query is stemmed.

We find it more instructive to test stemming separately against a stemming gold standard and assess the relevance of stemming for IR by testing the stemming gold standard on IR performance. If stemming turns out to be relevant for IR, then researchers should continue to develop stemming algorithms towards the gold standard. In the other case, one wonder whether IR-improving term conflation methods should be called stemmers.

In order to assess the cross-linguistic applicability of our stemming algorithm we have chosen languages spanning spectrum of morphological typology – from isolating to highly suffixing – Maori, English, Swedish and Kuku Yalanji (Dryer 2005). As training data we used only the set of words from a bible translation to emphasize the applicability to resource-scarce languages. Table 4.4 contains information on the bible versions used.

For these four languages we devised a stemming gold standard using Bauer et al. (1993); Williams (1971) for Maori and Patz (2002); Hershberger and Hershberger (1982) for Kuku Yalanji, languages not generally known to the author. So as not to let the test set be dominated by too many simple test cases, the selection of test set cases was done as follows:

1. Select a random word w_1 from W for the corresponding language

2. Select a random number i in $0 \leq i \leq |w_1| - 1$

3. Select a random word w_2 from the subset of words from $W \setminus \{w_1\}$ sharing i initial characters with w_1

4. Mark the pair w_1, w_2 to be of the same stem according to traditional linguistic analysis

	same-stem		diff.-stem	
Language	Correct	Total	Correct	Total
Maori	10	13	100	100
English	97	100	100	100
Swedish	96	100	100	100
Kuku Yalanji	94	100	100	100

Table 4.10: Evaluation results.

This was repeated until 200 pairs of words for each language had been selected, 100 same-stem and 100 not same-stem. Except for Maori where we could only really find 13 same-stem cases this way, all involving active-passive alternating verbs (described in detail in Sanders (1990)).

The evaluation results are shown in table 4.10.

The errors fall into just one major type, in which the algorithm is too cautious to conflate; it is when two words do share the stem but where one of the suffixes is rather uncommon (possible because it is really composite) and therefore it is not in the grow-set of the other suffix; for example Swedish *skap-ade-s* (passive past) and *skap-are-n-s* (agent noun genitive definite). We also expected false positives in the form of random resemblances involing short words and short affixes; e.g., *as* versus *a* but no such cases seem to have occurred in the test set in any of the languages.

We have done attempted a comparison with other existing stemmers, mainly because they tend not be aimed at an open set of languages and those which are, are really not fully supervised and we fear we might not do justice to them in setting parameters (see Related Work section). The widely known Porter stemmer (Porter 1980) for English scores exactly the same result for English as our stemmer, which suggests than an unsupervised approach may come very close to explicitly human-informed stemmers. Many other stemmers, however, are superior to ours in the sense that they can stem a single word correctly whereas ours requires a pair of words to make a decision. This is especially relevant when large bodies of data needs to be stem-indexed as it would take quadratic time (in the number of words) in our setting.

4.5 Related Work

A full survey of stemming algorithms for specific languages or languages like English has more or less fully been done elsewhere (the technology becoming relatively mature cf. Erjavec and Džeroski (2004); Frakes and Fox (2003); Goldsmith et al. (2001); Melucci and Orio (2003); Rogati et al. (2003); Hull (1996); Galambos (2004); Flenner (1994) and references therein). We will focus instead on unsupervised approaches for a wider class of languages.

Melucci and Orio (2003) present a very elegant unsupervised stemming model. While training does not require any manually annotated data, some architectural choices depending on the language still has to be supplied by a human. If this can be overcome in an easy way, it would be very interesting to test their Baum-Welch training approach versus the explicit heuristics in this paper, especially on a wider scope of languages than given in their paper. The unsupervised stemmer outlined in Goldsmith et al. (2001) actually requires a lot of parameters to be tweaked humanly and mainly targets languages with one-slot morphology.

Other systems for unsupervised learning of morphology which do not explicitly do stemming could easily be transformed into stemmer. Work includes Jacquemin (1997); Yarowsky and Wicentowski (2000); Baroni et al. (2002); Clark (2001a); Ćavar et al. (2004); Brent et al. (1995); Déjean (1998a); Snover et al. (2002); Argamon et al. (2004); Goldsmith (2001); Neuvel and Fulop (2002); Gaussier (1999); Sharma et al. (2002); Oliver (2004) and other articles by the same authors. All of these systems, however, require some parameter tweaking as it is and perhaps one more if transformed to stemmers, so there is still work missing before they can be compared on equal grounds to the stemmer described here. Given that they use essentially the same kind of evidence, it is likely that some of them, especially Creutz and Lagus (2006), will reach just as competitive results on the same task.

4.6 Conclusion

We have presented a fully unsupervised human-intervention-free algorithm for stemming for an open class of languages showing very promising accuracy results. Since it does not rely on existing large data collections or other linguistic resources than raw text it is especially attractive for low-density languages. Although polynomial in time, it appears rather slow in practice and may not be suitable for stemming huge text collections. Future directions include investigating whether there is a speedier shortcut and better, more systematic, approach to layered morphology i.e., languages which allow affixes to be stacked.

4.7 Acknowledgements

The author has benefited much from discussions with Bengt Nordström. We also wish to extend special thanks to ASEDA for granting access to electronic versions of the Kuku Yalanji bible texts.

Chapter 5

Paper IV: A Fine-Grained Model of Language Identification

Harald Hammarström
Department of Computing Science
Chalmers University of Technology
harald2@cs.chalmers.se

Abstract

Existing state-of-the-art techniques to identify the language of a written text most often use a 3-gram frequency table as basis for 'fingerprinting' a language. While this approach performs very well in practice (99%-ish accuracy) if the text to be classified is of size, say, 100 characters or more, it cannot be used reliably to classify even shorter input, nor can it detect if the input is a concatenation of text from several languages. The present paper describes a more fine-grained model which aims at reliable classification of input as short as one word. It is heavier than the classic classifiers in that it stores a large frequency dictionary as well as an affix table, but with significant gains in elegance since the classifier is entirely unsupervised. Classifying a short input query in multilingual information retrieval is the target application for which the method was developed, but also tools such as spell-checkers will benefit from recognising occasional interspersed foreign words. It is also acknowledged that a lot of practical applications do not need this fine level of granularity, and thus remain largely unbenefited by the new model. Not having access to real-world multi-lingual query data, we evaluate rigorously, using a 32-language parallel bible corpus, that

accuracy is competitive on short input as well as multi-lingual input, and not only for a set of European languages with similar morphological typology.

5.1 Introduction

The language identification problem is to decide for a natural language text which language it is written in. The usual setting is to assume that one has access to training corpora beforehand for the languages to be considered. Some language fingerprint model is built from the training corpora and then classification of unseen text (belonging to one of the languages at hand) is performed through this model.

Existing state-of-the-art techniques rely on a surprisingly simple model, namely, a frequency table of character 3-grams for each language, read off directly from the training corpora. The corresponding 3-gram frequency table for the text to be classified is then compared to each stored language by some rank-frequency metric. In practice, this approach performs very well (99%-ish accuracy) if the text to be classified is of size, say, 100 characters or more (Juola 2006). Thus the language identification problem is a solved problem for most practical applications.

However, the crude 3-character gram method has a certain drawback (which may or may not be practical problem), in that it is not monotone. That is, if two texts s_1, s_2 are classified as l_1, l_2 respectively, then it is not certain that the concatenation of s_1 and s_2 is classified as either l_1 or l_2.

We will present an alternative model which aims at reliable classification of new text as short as one word. This model combines a frequency dictionary from each training corpus and a component that tries to recognize completely unseen words by looking at affixes (which would e.g., identify a word like *jihading* 'fighting the jihad' correctly as English). This latter component is crucial, not only for languages which make more use of affixes than English, but because there will always pop up completely novel words for any natural language no matter what size the training data. The affix detection technique implemented also builds from the same training corpora and requires no extra supervision or work by a human.

There are certainly practical applications which do require reliable classification of small segments and autodetection of language switches. These include spell checkers that wish to disregard interspersed foreign words, text-to-speech systems that make intermediate use of grapheme-to-phoneme conversion likewise wish to indentify interspersed foreign words, and multilingual information retrieval systems would benefit from knowing the language(s) of the words of a short query. For a lot of other practical applications, the granularity of the proposed new model is superfluous. For these applications, the only advantage of the proposed model is elegance

and absolute lack of training supervision.

The resultant language identifier is evaluated using bible corpora for 32 languages, spanning the full range of morphological typology of languages of the world (Dryer 2005). Both its ability to classify short segments into one language and to autodetect short segments that may be composed of several languages, are evaluated. However, we do not compare these figures to existing systems, because they were not designed for classifying short segments accurately (and thus perform very poorly)[1]. On longer segments, i.e., 100 characters, performance is near perfect, and it is presumed that the state-of-the-art systems would also perform near perfect if tested on the same set.

With the improved accuracy on short segments and wide typological testing range, we hope to have met the challenges for written language identification set out in a recent survey article by (Hughes et al. 2006).

All the training corpora used in this paper are bible corpora, since they are the only sufficiently large corpora available for a reasonably varied set of languages.

5.2 Previous Work

My full bibliography of works dealing narrowly with written language identification spans over 100 articles, a handful of technical reports and one PhD thesis (Ziegler 1991) – it is therefore not possible to review them all here. Many pointers to older work and language identification of speech signals are given in Muthusamy and Spitz (1997); Caseiro (1999). Sibun and Reynar (1996) is an excellent review and comparison of techniques used in early work.

For the language identification problem in the setting as in this paper, namely, written language identification trained on reference language data, two different feature models have been prevalent. One that looks at common words and one based on character n-grams (Grefenstette 1995; Cavnar and Trenkle 1994; Damashek 1995; Dunning 1994) – see Martin et al. (2006); Kruengkrai et al. (2005) for refinements of the n. The classification can then be done by comparing input text features to reference language features using rank-order statistics. More recent work in this direction has aimed at trimming overweight feature models (Poutsma 2002; Takci and Sogukpinar 2004) or at combining n-gram and whole word features (Prager 2000). See, however Biemann and Teresniak (2005) for a novel, completely different approach based on words clustered on sentence-co-occurrence. (The accu-

[1]There would also have been practical problems in doing justice as many descriptions of existing systems hide information on parameter tweaking. Online systems we have found do not allow uploading the training/test set we use, which is crucial in order to assess language-dependentness.

racy of this identifier is comparable to the older approaches, but it is not, as claimed therein, unsupervised, because there is a very large number of manually set parameters/thresholds and word-frequency statistics are gathered from curated corpora.) There is also more recent work targeting web pages specifically (Xafopoulos et al. 2004; Martins and Silva 2005; Lins and Gonçalves 2004), that address the proper treatment of HTML tags.

Whereas the language identification problem has variously been labelled 'easy' and 'solved' (McNamee 2005), it depends on whether one sets the goal higher than distinguishing non-minimal noise-free samples of European languages. Some recent articles (Murthy and Kumar 2006; da Silva and Lopes 2006b:a) identify practical problems where this is not so. For instance, as far as we can ascertain, the best systems in van Noord's Online Summary[2] minimally require some 20 characters of text to make a judgment at all. Nor are they capable of realizing that a sample text is a concatenation of two languages. For example, The Xerox MLTT Language Identifier[3] classifies the sentence 'good fish prefer their snake' correctly as English, the sentence 'fina fiskar sprattlar inte ofta' correctly as Swedish, but the concatenation of the two is classified as Norwegian (even though there is actually no legal Norwegian word in either sentence).

As indicated already, the present method seeks to tackle also smaller sample texts, which is crucial in order to be able to track whether a text is a composition of words from several languages. While the classic n-gram approaches have found that a good $n = 3$, i.e., that salient morphemes can be approximated as being exactly 3 characters, a more elegant alternative is to hold this variable, so that salient affixes can have any length in any language. Furthermore, we wish to extend the testing scope, as present published testing has been only on a rather small set of European languages.

5.3 Definitions and Preliminaries

Start with a finite non-empty alphabet Σ. The following terminology and notation will be used.

word: a non-empty finite string over Σ. Thus the set of all possible words can be denoted Σ^+. Lowercase w with subscripts will be used for variables over words. A word will be enclosed in quotes if confusion could arise otherwise.

sentence: a finite non-empty tuple of words $\langle w_1, w_2, \ldots, w_n \rangle$. Commas and brackets will be omitted when no confusion can arise. However,

[2] `http://odur.let.rug.nl/~vannoord/TextCat/competitors.html` accessed the 25th of May 2005.

[3] `http://www.xrce.xerox.com/competencies/content-analysis/tools/guesser` accessed 20 Jan 2007.

variables that range over tuples, e.g., $\langle l \rangle$, will always be written with brackets.

S_Σ : let $S_\Sigma = \{\langle w_1 w_2 \dots w_n \rangle | w_i \in \Sigma^+, n \in \mathbf{N}\}$ denote the set of all possible sentences.

language: a probability distribution over sentences $L : S_\Sigma \to [0, 1]$ such that $\sum_{\langle s \rangle} L(s) = 1$.

training corpus: a finite sequence of sentences. However, we will never make use of the order of sentences, or order or words in the sentences, so a training corpus may be equated with its bag of words. Thus, if T is a training corpus, let $f_T(w)$ denote the frequency of the word w in T. Also, use $W_T = \{w | f_T(w) \geq 1\}$ for the *set* of words in the training corpus.

names and variables: Unless we are talking about existing natural languages, e.g., English, natural numbers $1, 2, \dots$ will be used for language names. $\Sigma_1, \Sigma_2, \dots$ will be used for their corresponding alphabets, with $\Sigma = \bigcup_i \Sigma_i$ for the mother alphabet. $L_1, L_2, \dots$ will be used for languages, i.e., probability distributions, and coindexed $T_1, T_2, \dots$ for training corpora (where T_i is assumed to be sampled from L_i).

The idea is of course that sentences which are illegal or ill-formed in some natural language will have zero probability and legal sentences will have a non-zero probability corresponding to their relative frequency. A natural way to see how a natural language should correspond to such a formal probabilistic language is to consider ever increasing amounts of natural language text and let the probability of each sentence be its limiting relative frequency. This correspondence requires that this limit actually exists for all sentences. If there are natural languages that do not live up to this, or which cannot be modelled so with an acceptable level of discrepancy, they should not be thought of as languages in our terminology.

Our notion of language is a generalization of the more common formalization of natural language as a *set* of sentences. We actually need this greater flexibility in order for language identifiers to exploit the fact that some words (and thus some sentences) which are legal in several natural languages may be distinguished by their different levels of frequency. It also provides a framework for gracious treatment of new words and proper names which are so ubiquitous in open domain natural language text (such as newspaper text) that they cannot be "abstracted away". With the probability model we have the power to say that any word is possible in any language, for example as a proper name, but it is more probable that an instance of e.g., 'the' is from English than in some other language where it may have occurred as a proper name.

5.4 A Fine-Grained Model of Language Identification

From the input of a training corpus, the proposed model characterizes a language using the following two components:

Frequency dictionary: Stores each seen word and its (relative) frequency. The frequency of seen words is a very powerful predictor of a language.

Unsupervised affix detection: Salient affixes are extracted (in an unsupervised manner), which form the basis for a probabilistic guessing of previously unseen words.

These two components are combined into a *word emission probability* distribution that aims to predict how likely a language is to have emitted a given word. In principle, a collection of such probability distributions are sufficient to make up a standard case of language identifier that always outputs exactly one language. However, we shall also use another component, a *language holdback bias*, to enable intuitively correct identification of text that is concatenated from several languages.

5.4.1 Word Emission Probability

A frequency dictionary FD_l is built simply as:

$$FD_l(w) = \frac{f_{T_l}(w)}{\sum_{w' \in \Sigma} f_{T_l}(w')}$$

Following (Hammarström 2006a) we use an unsupervised algorithm to gather information on the salient affixes for a given language. The algorithm uses W_l as its input and outputs a probability distribution on character strings that aims to say whether a given segment is likely to be a characteristic prefix or suffix for the language at hand. To be more precise, the probability distribution aims to capture the notion of morpheme probability that one arrives at if: 1. A linguist does a morphemic segmentation of the word types (not words tokens) occurring in a corpus, 2. The frequencies of the individual morphemes, in prefix or suffix position, are interpreted as probabilities. For example, *-qvj* would likely get zero probability in an English corpus. An example output, adapted from Hammarström (2006a), is given in Table 5.1, sorted on highest probability. The outcome of the algorithm for languages which do not have any morphology at all is a fairly even spread of probability mass over initial and final characters of the words of the language in question. For reasons of space, the reader is referred to the said paper for a discussion of the inner workings and alternative algorithms.

As mentioned, the output from the affix extraction is a probability distribution over affixes. What we need is a probability distribution over words,

Table 5.1: Comparative figures for prefix vs. suffix detection for three sample languages.

Swedish		English		Swahili	
för-	0.097	*-ed*	0.132	*-a*	0.100
-en	0.086	*-eth*	0.109	*wa-*	0.095
-na	0.036	*-iah*	0.099	*ali-*	0.065
-ade	0.035	*-ly*	0.090	*nita-*	0.059
-a	0.034	*-ings*	0.068	*aka-*	0.049
-ar	0.033	*-ing*	0.062	*ni-*	0.046
-er	0.033	*-ity*	0.059	*ku-*	0.044
-as	0.032	*-edst*	0.058	*ata-*	0.042
-s	0.031	*-ites*	0.046	*ha-*	0.032
-de	0.031	*-s'*	0.036	*a-*	0.031
...	...	...	...	...	...

in which any word ending in some salient suffix should have nonzero probability. One quite reasonable way to achieve this is to assign geometrically decreasing probabilities for longer and longer words. Thinking in this way, we let all observed (in W_l) word lengths get the probability mass proportional to the number of observed words with such lengths, and unseen word lengths get geometrically decreasing probability. Thus, to get a well-defined probability distribution over words based on the affix probability distribution, we multiply together the word-length mass for w with the highest (not necessarily longest!) matching, if any, affix probability, for a given word w. The details aren't interesting, but use $A_l(w)$ to denote the just described affix-based probability distribution.

Putting the affix detection together with the frequency dictionary to make an emission probability involves a related kind of estimate. How much probability mass should be assigned to seen vs. unseen words? There are probably many similar alternatives, but here we have simply guessed that unseen words are like hapax words, and assigned the probability mass proportions to be like the proportion of hapax words: $\alpha_l = \frac{|\{w \in W_l | f_{T_l}(w)=1\}|}{|W_l|}$.

We are now ready to define emission probability:

$$P_l(w) = \begin{cases} (1-\alpha_l) \cdot FD_l(w) & \text{if } w \in W_l \\ \alpha_l \cdot A_l(w) & \text{if } w \notin W_l \end{cases}$$

It can happen that there is more mass given to an unseen word than to a (rare) seen word, even within one particular language. In fact, proportions vary quite wildly between languages, as can be seen in Table 5.2 with figures computed on the translations of the same bible text.

Table 5.2: Some indications as to the widely differing identification cues for three languages; the polysynthetic Greenlandic versus the almost isolating Haitian creole.

Language	$\|T\|$	$\|W\|$	α	$argmax_w(FD(w))$	
Greenlandic	382188	107918	0.706	*taava* (then)	0.00857
Swedish	758773	26825	0.407	*och* (and)	0.05566
Haitian creole	904915	7796	0.335	*yo* (PL/they)	0.05531

5.4.2 Language Holdback Bias

If we have $L_1, \ldots, L_n$ languages, the previous section shows how to construct the corresponding $P_1, \ldots, P_n$ probability distributions over words. Next, we shall define a family of probability measures over *sequences of words*. There will be one probability distribution for each language tuple of the same length as the sequence to be measured:

$$P_{l_1 l_2 \ldots l_m}(w_1 w_2 \ldots w_m) = \prod_i P_{l_i}(w_i)$$

Given a sequence of words we could then naïvely decide which language(s) it most probably belonged to by listing each tuple of the appropriate length and computing which tuple has the highest probability of having generated the sequence of words. However, for several reasons, such an approach is not advisable. First, with n languages there are n^m language tuples so it would not be tractable to enumerate them all. Second, the probability measures so defined, the output will be the concatenation of the most probable language for each word individually. This is probably not what we want since many words that are legal in several languages differ in frequency. Consider a sequence of a million words indisputably belonging to language L_1, and, interspersed inside, a word that is legal in both L_1 and L_2 but slightly more common in L_2. The naïve language identifier would yield L_2 disregarding the suggestive surrounding million words of L_1. While it is technically not impossible that it is a concatenation of the two languages, a human would never see it as that. Third, it's not clear how to see if an input sequence is non-trivially legal in more than one way (i.e., there are several satisfactory language tuples). Either we insert some kind of threshold which would be hard to know how to set, or we have to say that pretty much all tuples are satisfactory identification of the sequence only with some degree variation.

For the first problem, it is easy to see that not all tuples need to be enumerated to get the maximally probable one (if we want only this one, rather than the probabilities for all). As defined, the emission probabilities depend only on a particular word, not anything else in the sequence, so maximas can be computed locally in the sequence and glued together as in any standard application of dynamic programming. For the second and

third problem, we shall propose a refinement of the strategy that obviates the need for any thresholds.

We propose that a machine language identifier like ours should have a *bias* towards minimizing the number of times we change languages in an identification sequence. To be more precise, the prior probability that a sequence should switch language c times should decrease exponentially in c. Also, other things being equal, the longer the sequence the stronger the bias should be, i.e., it should not be less likely that a million word sequence should switch language once somewhere within it, than that a two-word sequence should switch language (once) within it. This is the way to say that having seen a million words of language L_1 counts for more than having seen just one word of L_1. We do not see any basis for this to be a sequential property, e.g., that language switches are significantly more (or less) likely after or before certain words, wherefore a (H)MM-modeling technique offers no advantage.

Formally, let $C(l_1 l_2 \ldots l_m) = |\{i | l_i \neq l_{i+1}\}|$ denote the number of times a change in language occurs in a language sequence. Clearly, we have $0 \leq c \leq m-1$. Let $\langle l \rangle = l_1 l_2 \ldots l_m$ be an arbitrary language tuple under consideration and $c = C(\langle l \rangle)$ its number of switches. Now, for any language identifier parametrized on c and m, we wish the bias, regardless of the particular languages at hand, to ensure that:

$$\frac{P(c,m)}{P(c+k,m)} \geq 2^k \qquad \text{for all } k \geq 0, m$$

$$P(c,m) > P(c,m+k) \qquad \text{for all } k \geq 1, c$$

A simple fulfilment of these is the following **Language Holdback Bias** function $B(c,m)$:

$$B(c,m) = \frac{1}{m^c} \cdot \frac{1}{\sum_{0 \leq i \leq m-1} \frac{1}{m^i}}$$

There of course alternative bias functions that also fulfil the desiderata, but this is the simplest one. Now, with the bias function defined we are ready to present our full definition of the output of the now rather sophisticated language identifier.

$$ID(w_1 \ldots w_m) = \begin{array}{l} \text{the set of all tuples } \langle l \rangle = l_1 \ldots l_m \\ \text{such that for all } \langle l' \rangle \\ B(C(\langle l \rangle), m) \cdot P_{\langle l \rangle}(w_1 \ldots w_m) \geq \\ B(C(\langle l' \rangle), m) \cdot P_{\langle l' \rangle}(w_1 \ldots w_m) \end{array}$$

The formula conveys the following: look for tuples with as few cuts (i.e., minimal c) as possible, that are such that they have higher probability, the bias respected, than any other tuple with *more* cuts. This is the key

feature which eliminates the need for a threshold. Thus, for example, a word sequence will be said to be of language L_l iff it has higher probability than any division of the sequence into two parts of different languages (or three parts etc). There may be several such languages, but hardly all, so the yield will be a strong prediction.

The following more procedural reformulation of the identification function may be easier to understand. It should also make it clear that language identification is still polynomial in the sequence length, since there are still no dependencies between the word-probabilities.

1. Find minimal c such that there exists a tuple $\langle l \rangle$ with $C(\langle l \rangle) = c$ and:

$$B(c, m) \cdot P_{\langle l \rangle}(w_1 \dots w_m) \geq$$
$$B(C(\langle l' \rangle), m) \cdot P_{\langle l' \rangle}(w_1 \dots w_m)$$
$$\text{for all } \langle l' \rangle \text{ with } C(\langle l \rangle) > c$$

2. Output all tuples $\langle l \rangle$ with $C(\langle l \rangle) = c$ and:

$$B(c, m) \cdot P_{\langle l \rangle}(w_1 \dots w_m) \geq$$
$$B(C(\langle l' \rangle), m) \cdot P_{\langle l' \rangle}(w_1 \dots w_m)$$
$$\text{for all } \langle l' \rangle \text{ with } C(\langle l \rangle) > c$$

5.4.3 Examples

Example 1: The kings hon walikusoma

Consider the sequence *the kings hon walikusoma* which consists of *the*, which is of course the English definite article; *kings* is the well-known English lexical item which does occur in the training corpus – it also happens to end in *-s* which is a very common Swedish inflectional ending (but there is no lexical item 'king' or 'kings' in Swedish); *hon* is a Swedish personal pronoun, abundantly occurring in the Swedish training corpus; and *walikusoma* is a well formed Swahili word whose individual morphemes all individually occur abundantly in the Swahili training corpus – but the perfectly well-formed word 'walikusoma' does not occur in the training corpus (it would mean 'they read you').

The individual word-probabilities as well as a selection of the more interesting tuple-probabilities for the sequence as a whole, are shown in Table 5.3. As can be seen, the $P_{eng,eng,swe,swa}$ value beats all tuples with zero or one switches. It also happens to beat all tuples with three switches and it is the only such tuple. Therefore, in this case, the output will be exactly English, English, Swedish, Swahili.

Example 2: The kings are there

The complicated interaction seen in the previous example does not disturb the "normal" easy class of classifications. Table 5.4 shows the word-probabilities for the almost trivial sentence *the kings are there*. There is a

Table 5.3: Example 1: $P_l(w)$ for a set of languages and some interesting words, followed by a selection of the more interesting tuple-probabilities.

	'the'	'kings'	'hon'	'walikusoma'
English	0.051522	0.000286	0.000003	0.000004
Swedish	0.000002	0.000040	0.000916	0.000043
Swahili	0.000218	0.000000	0.000000	0.000317

All one-language tuples

$P_{eng,eng,eng,eng}$	1.350e-016
$P_{swe,swe,swe,swe}$	2.468e-018
$P_{swa,swa,swa,swa}$	1.878e-025

Some top one-switch tuples

$P_{eng,swe,swe,swe}$	2.034e-014
$P_{eng,eng,swe,swe}$	1.465e-013
$P_{eng,eng,eng,swa}$	3.008e-015

The top two-switch tuple

$P_{eng,eng,swe,swa}$	2.701e-013

certain zero-switch tuple which is way ahead of the others. As it also beats all one-switch tuples (and no other zero-switch tuple does), it will be the output of the identifier.

Example 3: De la

There are instances where there are several "winning" tuples, though informal tests show that this is not achieved very often. The sequence *de la* is very common to both Spanish and French. In English it is not common at all. In Swedish *de* is a personal pronoun so it enjoys a certain frequency, whereas *la* is not a word in (bible) Swedish. Similarly, *la* is a negator in

Table 5.4: Example 2: $P_l(w)$ for a set of languages and some words that are very easy to classify, followed by examples to indicate that the dominance of a certain zero-switch tuple over some others.

	'the'	'kings'	'are'	'there'
English	0.051522	0.000286	0.002812	0.002065
Swedish	0.000002	0.000040	0.000006	0.000035
Swahili	0.000218	0.000000	0.000004	0.000006

$P_{eng,eng,eng,eng}$	8.5467629403443202e-011
$P_{swe,swe,swe,swe}$	1.2961894211016589e-020
$P_{swa,swa,swa,swa}$	2.5363460513704776e-023
...	

Table 5.5: Example 3: $P_l(w)$ for a set of languages and two words, followed by a selection of the more interesting tuple-probabilities.

	'de'	'la'		
French	0.029172	0.016325	$P_{fre,fre}$	0.0003174886
English	0.000000	0.000000	$P_{spa,spa}$	0.0003227756
Swedish	0.008400	0.000001	$P_{spa,fre}$	0.0001844997
Swahili	0.000000	0.001517	...	...
Spanish	0.033905	0.014280		

Swahili and is therefore fairly frequent. Table 5.5 shows the relevant probabilities. The output will be only the tuples spa, spa and fre, fre, because tuples like swe, swa and spa, fre lose out because of the bias, favouring few switches.

5.5 Evaluation and Discussion

Three extensive tests were performed using a parallel corpus of the bible in 32 languages, which contains languages from the isolating Maori to the record holding polysynthetic Greenlandic (Dryer 2005). In order to get a sufficiently cross-language comparable evaluation, size and randomness were equalized between languages the following way. A random verse from each chapter was selected (there are 1209 chapters in the bible). This was done once for the whole language set. Of course, these verses were removed from the training data. A random word from each selected verse was selected. This word-selection was done separately for each language. For each language, we thus get a set of randomly selected words E_l. Though 1209 word-selections were made for each language, many selections happened to select the same word. Thus the size of the E_l-sets varied from 350 (for Maori) to 974 (for Greenlandic). The discrepancy is not disturbing. Words are not entities of the same kind across languages, but our classifier operates on the granularity of words, and the desiderata is an evaluation of 'accuracy per (randomly selected) word'. An alternative, e.g., selecting 1000 unique words of each language would have made interpretation of the result difficult, because for Maori, it is likely that most of the 1000 words would have been *seen* words, occurring in other verses, whereas the opposite is the case for Greenlandic.

If E is a set of tuples (possibly one-word tuples), drawn for language l, we define the accuracy $R_E(l)$ of a language identifier ID:

$$R_E(l) = \frac{|\{\langle x\rangle | ID(\langle x\rangle) = l \text{ and } \langle x\rangle \in E\}|}{|E|}$$

One-word classification: The R_{E_l} was calculated for each of the 32 languages. Since the input sequence is of length 1, there will never be any

cuts, so the language identifier was set to output the language with highest probability of having emitted the input word. The E_l-sets as defined above may contain words that are "impossible" predict where they were taken from, on the basis of the word alone. For example, let's say a word w is legal in two languages but much more common in l_1 than l_2. If it happened to be drawn from L_{l_2}, it is hard to see how this can be predicted. However, we computed figures on the possible influence of this issue, and it turned out to be minor. Therefore, the results in Table 5.6 stand, but could be adjusted upwards by very small percentages.

Verse classification: To check how accurate the identifier was on longer segments, we chose to test on segments of roughly the length of a verse. Verses, in fact, happen to be around 100 characters long on average. From the 1209 verses selected (as above), those 100 verses thereof whose number of characters were closest to the average verse length of that language, were selected for testing. Denoting these 100-verse sets by V_l, the verse-classification accuracy R_{V_l} was calculated. This score, as well as data on average verse length, can be seen in Table 5.6.

4-tuple multilingual classification: A set of 1000 mixed language 4-tuples were built from $E_1, \ldots, E_{32}$ as follows.

1. Pick a random language l and pick two random words from that E_l.
2. Precede it with a random word from a random language $E_{l'}$.
3. Add a random word from a random language $E_{l''}$ at the end.

The results of this test was 193 (**19.3%**) fully correctly identified tuples and 204 (**20.4%**) with exactly one word misclassified.

Some figures are low, not surprisingly for languages with a lot of morphology, but overall we hold the results are very reasonable given the exceedingly difficult test problems of one-word and multi-language classification. It is very easy to make mistakes on single words when there are so many languages in the pool – the results are much higher if the number of competing languages is halved.

Unfortunately, we cannot contrast the verse-test with figures from competing state-of-the-art systems, as none of the systems known to us give enough details (on thresholds and such) to reconstruct a fair version of the classifier.

A matter requiring further commentary is the use of a bias function to do the job a scalar threshold value does in related work. (Human language identifiers, having the ability to assess syntactic and semantic coherence,

Table 5.6: Accuracies for the one-word and verse tests plus average verse length in characters ($\overline{V}$).

Language	1-word	Verse	$\overline{V}$
Haitian Creole	0.839	1.00	101.79
Zarma	0.781	1.00	99.45
Kekchi	0.720	1.00	148.78
English	0.678	1.00	104.19
Maori	0.665	1.00	107.73
Hindi	0.607	1.00	119.50
Hausa	0.605	1.00	94.10
Afrikaans	0.594	1.00	103.34
Danish	0.580	1.00	89.30
Cebuano	0.573	1.00	129.48
Icelandic	0.550	1.00	95.58
Swedish	0.547	1.00	107.20
Adamawa Fulfulde	0.539	1.00	96.57
German	0.533	1.00	103.52
Albanian	0.523	1.00	114.80
Spanish	0.511	1.00	95.83
French	0.507	1.00	101.83
Swahili	0.494	1.00	105.03
Slovene	0.488	1.00	100.12
Polish	0.487	1.00	144.52
Portuguese	0.481	1.00	98.41
Esperanto	0.473	1.00	97.80
Italian	0.473	1.00	116.80
Catalan	0.450	1.00	109.70
Dutch	0.415	1.00	109.36
Lithuanian	0.396	1.00	104.99
Hungarian	0.386	1.00	102.10
Latin	0.366	0.99	112.54
Turkish	0.348	0.95	93.43
Finnish	0.345	0.99	107.88
Malayalam	0.276	0.88	128.65
Greenlandic	0.222	0.87	126.99

need not use either.) Conceptually the bias function employed is nothing other than a complex system of thresholds, in terms of growth behaviour (exponential, linear etc.) rather than scalar values. Arguably, this is an elegance improvement, although it comes with the cost of being harder to understand, compute and analyse. Also, in the experiments reported above, the bias function approach experimentally outperforms a simple systems of scalar threshold values. For example, through supervised training we have tried tuning one single threshold value for all experiments, one threshold value individually for each language, different threshold values for different classification tasks (i.e., one for multi-language classification and one for single language classification) and so on, resulting in generally lower accuracy on the same test set (obviously, there is little room for presenting and discussing figures from these tests here). Nevertheless, it remains possible that some other, yet undiscovered, system of scalar thresholds is superior to the bias function.

5.6 Conclusions

We have described a new model with considerable elegance for language identification on small, possibly mixed languages segments. We have also added significantly to the set of published evaluations of a language identification system with a balanced cross-language test. For larger input texts the new model has excellent accuracy, but it is bigger and slower in practice than the existing state-of-the-art systems.

Bibliography

Albright, A. C. (2002). *The Identification of Bases in Morphological Paradigms*. PhD thesis, University of California at Los Angeles.

American Bible Society (1988). *Turkish Bible*. American Bible Society, Tulsa, Oklahoma.

American Bible Society (1999). *Bib La*. American Bible Society. This edition from 2003.

Andreev, N. D., editor (1965). *Statistiko-kombinatornoe modelirovanie Yazykov*. Akademia Nauk SSSR, Moskva.

Arabsorkhi, M. and Shamsfard, M. (2006). Unsupervised discovery of persian morphemes. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics.

Argamon, S., Akiva, N., Amit, A., and Kapah, O. (2004). Efficient unsupervised recursive word segmentation using minimum description length. In *COLING-04, 22-29 August 2004, Geneva, Switzerland*.

Bacchin, M., Ferro, N., and Melucci, M. (2002). The effectiveness of a graph-based algorithm for stemming. In Lim, E. P., Foo, S., Khoo, C. S. G., Chen, H., Fox, E. A., Urs, S. R., and Thanos, C., editors, *ICADL '02: Proceedings of the 5th International Conference on Asian Digital Libraries*, volume 2555 of *Lecture Notes in Computer Science*, pages 117–128. Springer-Verlag, Berlin.

Bacchin, M., Ferro, N., and Melucci, M. (2005). A probabilistic model for stemmer generation. *Information Processing and Management*, 41(1):121–137.

Baroni, M. (2000). *Distributional Cues in Morpheme Discovery: A Computational Model and Empirical Evidence*. PhD thesis, University of California, Los Angeles.

Baroni, M. (2003). Distribution-driven morpheme discovery: A computational/experimental study. *Yearbook of Morphology*, pages 213–248.

Baroni, M., Matiasek, J., and Trost, H. (2002). Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL/SIGPHON-2002*, pages 48–57.

Bauer, W., Parker, W., and Evans, T. K. (1993). *Maori*. Descriptive Grammars. Routledge, London & New York.

Belkin, M. and Goldsmith, J. (2002). Using eigenvectors of the bigram graph to infer morpheme identity. In *Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 41–47, Philadelphia. Association for Computational Linguistics.

Bernhard, D. (2005). Segmentation morphologique à partir de corpus. In *Actes de TALN & RÉCITAL 2005*, volume 1, pages 555–564. ATALA, Dourdan, France.

Bernhard, D. (2006). *Apprentissage de connaissances morphologiques pour lacquisition automatique de ressources lexicales*. PhD thesis, Université Joseph Fourier – Grenoble I.

Bernhard, D. (2007). 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*, volume 1, pages 367–376. Toulouse, France.

Bickel, B., Banjade, G., Gaenszle, M., Lieven, E., Paudyal, N., Rai, I., Rai, M., Rai, N. K., and Stoll3, S. (2007). Free prefix ordering in chintang. *Language*, 83(1):43–73.

Biemann, C. and Teresniak, S. (2005). Disentangling from babylonian confusion - unsupervised language identification. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, volume 3406 of *Lecture Notes in Computer Science*, pages 773–784. Springer.

Blomqvist, J. and Jastrup, P. (1998). *Grekisk Grammatik: Graesk grammatik*. Akademisk Forlag, København, 2 edition.

Bordag, S. (2005). Unsupervised knowledge-free morpheme boundary detection. In *Proceedings of Recent Advances in Natural Language Processing 2005 (RANLP '05)*. Borovets, Bulgaria.

Borin, L. (1991). *The Automatic Induction of Morphological Regularities*. PhD thesis, University of Uppsala.

Borin, L. (1997). Parole-korpusen vid språkbanken, göteborgs universitet. `http://spraakbanken.gu.se` accessed the 11th of Febuary 2004. 20 million words.

Brent, M. R. (1999). An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–105.

Brent, M. R., Murthy, S., and Lundberg, A. (1995). Discovering morphemic suffixes: A case study in minimum description length induction. In *Fifth International Workshop on Artificial Intelligence and Statistics, Ft. Lauderdale, Florida*.

British and Foreign Bible Society (1953). *Maandiko matakatifu ya Mungu yaitwaya Biblia, yaani Agano la kale na Agano jipya, katika lugha ya Kiswahili*. British and Foreign Bible Society, London, England.

Caseiro, D. (1999). Automatic language identification bibliography. `http://www.phys.uni.torun.pl/kmk/projects/ali-bib.html` accessed the 25th of May 2005.

Ćavar, D., Herring, J., Ikuta, T., Rodrigues, P., and Schrementi, G. (2004). On induction of morphology grammars and its role in bootstrapping. In Jäger, G., Monachesi, P., Penn, G., and Wintner, S., editors, *Proceedings of Formal Grammar 2004*, pages 47–62.

Ćavar, D., Rodrigues, P., and Schrementi, G. (2005). Unsupervised morphology induction for part-of-speech tagging. *U. Penn Working Papers in Linguistics*, 10(1).

Cavnar, W. B. and Trenkle, J. M. (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US.

Chan, E. (2006). Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology at HLT-NAACL 2006*, pages 69–78. Association for Computational Linguistics, New York City, USA.

Clark, A. (2001a). Learning morphology with pair hidden markov models. In *ACL (Companion Volume)*, pages 55–60.

Clark, A. (2001b). Partially supervised learning of morphology with stochastic transducers. In *Proceedings of Natural Language Processing Pacific Rim Symposium, NLPRS 2001*, pages 341–348, Tokyo, Japan.

Clark, A. (2002). Memory-based learning of morphology with stochastic transducers. In *Proceedings of the ACL 2002*. Association for Computational Linguistics, Philadephia.

Creutz, M. (2003). Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the ACL 2003*, pages 280–287. Sapporo, Japan.

Creutz, M. (2006). *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*. PhD thesis, Helsinki University of Technology, Espoo, Finland.

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON), Philadelphia, July 2002*, pages 21–30. Association for Computational Linguistics.

Creutz, M. and Lagus, K. (2004). 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)*, pages 43–51. Barcelona.

Creutz, M. and Lagus, K. (2005a). 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 '05), 15-17 June, Espoo, Finland*, pages 106–113. Espoo.

Creutz, M. and Lagus, K. (2005b). Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. Technical report, Publications in Computer and Information Science, Report A81, Helsinki University of Technology.

Creutz, M. and Lagus, K. (2006). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, pages 1–33.

Creutz, M., Lagus, K., Lindén, K., and Virpioja, S. (2005a). Morfessor and hutmegs: Unsupervised morpheme segmentation for highly-inflecting and compounding languages. In *Proceedings of the Second Baltic Conference on Human Language Technologies, Tallinn, 4 - 5 April*, pages 107–112. Tallinn, Estonia.

Creutz, M., Lagus, K., and Virpioja, S. (2005b). Unsupervised morphology induction using morfessor. In Yli-Jyrä, A., Karttunen, L., and Karhumäki, J., editors, *Finite State Methods in Natural Language Processing: 5th*

International Workshop, FSMNLP 2005, Helsinki, Finland, September 1-2, 2005. Revised Papers, volume 4002 of *Lecture Notes in Computer Science*, pages 300–301. Springer-Verlag, Berlin.

Creutz, M. and Lindén, K. (2004). Morpheme segmentation gold standards for Finnish and English. publications in computer and information science, report a77, helsinki university of technology. Technical report, Publications in Computer and Information Science, Report A77, Helsinki University of Technology.

da Silva, J. F. and Lopes, G. P. (2006a). Identification of document language is not yet a completely solved problem. In *CIMCA '06: Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, pages 212–219, Washington, DC, USA. IEEE Computer Society.

da Silva, J. F. and Lopes, J. G. P. (2006b). Identification of document language in hard contexts. In *Proceedings of the SIGIR 2006 Workshop on New Directions in Multilingual Information Access, Seattle, USA*.

Damashek, M. (1995). Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *Science*, 267(5199):843–848.

Dasgupta, S. and Ng, V. (2007). High-performance, language-independent morphological segmentation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 155–163, Rochester, New York. Association for Computational Linguistics.

de Kock, J. and Bossaert, W. (1974). *Introducción a la lingüística automática en las lenguas Románicas*, volume 202 of *Biblioteca románica hispánica 2: Estudios y ensayos*. Gredos, Madrid.

de Kock, J. and Bossaert, W. (1978). *The Morpheme: An Experiment in Quantitative and Computational Linguistics*. Van Gorcum, Amsterdam.

Déjean, H. (1998a). *Concepts et algorithmes pour la découverte des structures formelles des langues*. PhD thesis, Université de Caen Basse Normandie.

Déjean, H. (1998b). Morphemes as a necessary concept for structures discovery from untagged corpora. In *NeMLaP3/CoNLL98 Workshop on Paradigms and Grounding in Language Learning*, pages 295–298. Association for Computational Linguistics, Philadephia.

Demberg, V. (2007). A language-independent unsupervised model for morphological segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 920–927, Prague, Czech Republic. Association for Computational Linguistics.

Dixon, R. M. W. and Aikhenvald, A. (2002a). Introduction. In Dixon, R. M. W. and Aikhenvald, A., editors, *Word: A Cross-linguistic Typology*, pages 1–41. Cambridge University Press.

Dixon, R. M. W. and Aikhenvald, A., editors (2002b). *Word: A Cross-linguistic Typology*. Cambridge University Press.

Dryer, M. S. (2005). Prefixing versus suffixing in inflectional morphology. In Comrie, B., Dryer, M. S., Gil, D., and Haspelmath, M., editors, *World Atlas of Language Structures*, pages 110–113. Oxford University Press.

Dunning, T. (1994). Statistical identification of language. Technical report, Techical Report MCCS-94-273, Computing Research Lab (CRL), New Mexico State University.

Džeroski, S. and Erjavec, T. (1997). Learning slovene declensions with foidl. In Daelemans, W., Weijters, T., and van der Bosch, A., editors, *ECML'97 – Workshop Notes on Empirical Learning of Natural Language Tasks*, pages 49–60, Prague. University of Economics.

Džeroski, S. and Erjavec, T. (2000). Learning to lemmatise slovene words. In Cussens, J. and Dzeroski, S., editors, *Learning Language in Logic*, volume 1925 of *Lecture Notes in Computer Science*, pages 69–88. Springer-Verlag, Berlin.

Erjavec, T. and Džeroski, S. (2004). Machine learning of morphosyntactic structure: Lemmatizing slovene words. *Applied Artificial Intelligence*, 18:17–41.

Flenner, G. (1992). *Ein quantitatives Morphsegmentierungsverfahren für spanische Wortformen*. PhD thesis, Georg-August-Universität Göttingen.

Flenner, G. (1994). Ein quantitatives morphsegmentierungssystem für spanische wortformen. In Klenk, U., editor, *Computatio Linguae II: Aufsätze zur algorithmischen und Quantitativen Analyse der Sprache*, volume 83 of *Zeitschrift für Dialektologie und Linguistik: Beihefte*, pages 31–62. Franz Steiner, Stuttgart.

Flenner, G. (1995). Quantitative morphsegmentierung im spanischen auf phonologischer basis. *Sprache und Datenverarbeitung*, 19(2):63–78.

Forsberg, M. (2007). *Three Tools for Language Processing: BNF Converter, Functional Morphology, and Extract*. PhD thesis, Chalmers University of Technology, Gothenburg.

Forsberg, M., Hammarström, H., and Ranta, A. (2006). Lexicon extraction from raw text data. In Salakoski, T., Ginter, F., Pyysalo, S., and Pahikkala, T., editors, *Advances in Natural Language Processing: Proceedings of the 5th International Conference, FinTAL 2006 Turku, Finland, August 23-25, 2006*, volume 4139 of *Lecture Notes in Computer Science*, pages 488–499. Springer-Verlag, Berlin.

Frakes, W. B. and Fox, C. J. (2003). Strength and similarity of affix removal stemming algorithms. *SIGIR Forum*, 37(1):26–30.

Francis, N. W. and Kucera, H. (1964). Brown corpus. Department of Linguistics, Brown University, Providence, Rhode Island. 1 million words.

Galambos, L. (2004). *Multilingual Stemmer in Web Environment*. PhD thesis, Faculty of Mathematics and Physics, Charles University in Prague.

Gammon, E. (1969). Quantitative approximations to the word. In *International conference on computational linguistics, COLING, 1-4 September 1969, Sånga-Säby, Sweden*, pages 1–28. Stockholm.

Gaussier, É. (1999). Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-1999)*. Association for Computational Linguistics, Philadephia.

Gelbukh, A. F., Alexandrov, M., and Han, S.-Y. (2004). Detecting inflection patterns in natural language by minimization of morphological model. In Sanfeliu, A., Trinidad, J. F. M., and Carrasco-Ochoa, J. A., editors, *Proceedings of Progress in Pattern Recognition, Image Analysis and Applications, 9th Iberoamerican Congress on Pattern Recognition, CIARP '04*, volume 3287 of *Lecture Notes in Computer Science*, pages 432–438. Springer-Verlag, Berlin.

Gelbukh, A. F. and Sidorov, G. (2003). Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing, 4th International Conference, CICLing 2003, Mexico City, Mexico, February 16-22, 2003, Proceedings*, volume 2588 of *Lecture Notes in Computer Science*, pages 215–220. Springer-Verlag, Berlin.

Goldsmith, J. (2000). Linguistica: An automatic morphological analyzer. In Okrent, A. and Boyle, J., editors, *Proceedings from the Main Session of the Chicago Linguistic Society's thirty-sixth Meeting*, pages 125–139. Chicago Linguistics Society, Chicago.

Goldsmith, J. (2001). Unsupervised learning of the morphology of natural language. *Computational Linguistics*, 27(2):153–198.

Goldsmith, J. (2004). An algorithm for the unsupervised learning of morphology. Manuscript.

Goldsmith, J., Higgins, D., and Soglasnova, S. (2001). Automatic language-specific stemming in information retrieval. In Peters, C., editor, *Cross-Language Information Retrieval and Evaluation: Proceedings of the CLEF 2000 Workshop*, Lecture Notes in Computer Science, pages 273–283. Springer-Verlag, Berlin.

Goldsmith, J. and Hu, Y. (2004). From signatures to finite state automata. Technical report TR-2005-05, Department of Computer Science, University of Chicago.

Goldsmith, J., Hu, Y., Matveeva, I., and Sprague, C. (2005). A heuristic for morpheme discovery based on string edit distance. Techincal Report of Computer Science Department, University of Chicago.

Goldsmith, J. and O'Brien, J. (2007). Learning inflectional classes. *Language Learning and Development*, 24(4):219–250.

Goldsmith, J. A. (2006). An algorithm for the unsupervised learning of morphology. *TODOComputational Linguistics*, 12(4):353–371.

Grefenstette, G. (1995). Comparing two language identification schemes. In Bolasco, S., Lebart, L., and Salem, A., editors, *The proceedings of 3rd International Conference on Statistical Analysis of Textual Data (JADT 95), Rome, Italy, Dec. 1995.*

Hafer, M. A. and Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information and Storge Retrieval*, 10:371–385.

Hall, A. T. (1999). The phonological word: A review. In Hall, A. T. and Kleinheiz, U., editors, *Studies on the Phonological Word*, volume 174 of *Current Issues in Linguistic Theory*, pages 1–22. John Benjamins, Amsterdam.

Hammarström, H. (2005). A new algorithm for unsupervised induction of concatenative morphology. In Yli-Jyrä, A., Karttunen, L., and Karhumäki, J., editors, *Finite State Methods in Natural Language Processing: 5th International Workshop, FSMNLP 2005, Helsinki, Finland, September 1-2, 2005. Revised Papers*, volume 4002 of *Lecture Notes in Computer Science*, pages 288–289. Springer-Verlag, Berlin.

Hammarström, H. (2006a). A naive theory of morphology and an algorithm for extraction. In Wicentowski, R. and Kondrak, G., editors, *SIGPHON 2006: Eighth Meeting of the Proceedings of the ACL Special Interest Group on Computational Phonology, 8 June 2006, New*

York City, USA, pages 79–88. Association for Computational Linguistics. `http://www.cs.chalmers.se/~harald2/sigphon06.pdf`.

Hammarström, H. (2006b). Poor man's stemming: Unsupervised recognition of same-stem words. In Ng, H. T., Leong, M.-K., Kan, M.-Y., and Ji, D., editors, *Information Retrieval Technology: Proceedings of the Third Asia Information retrieval Symposium, AIRS 2006, Singapore, October 2006*, volume 4182 of *Lecture Notes in Computer Science*, pages 323–337. Springer-Verlag, Berlin.

Hammarström, H. (2007). A fine-grained model for language identification. In *Proceedings of iNEWS-07 Workshop at SIGIR 2007, 23-27 July 2007, Amsterdam*, pages 14–20. ACM.

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

Harris, Z. S. (1970). Morpheme boundaries within words: Report on a computer test. In Harris, Z. S., editor, *Papers in Structural and Transformational Linguistics*, volume 1 of *Formal Linguistics Series*, pages 68–77. D. Reidel, Dordrecht.

Hathout, N. (2005). Exploiter la structure analogique du lexique construit: une approche computationelle. *Cahiers de Lexicologie*, 87(2):1–24.

Haywood, J. A. and Nahmad, H. M. (1962). *New Arabic Grammar of the Written Language*. Harvard University Press, 2 edition.

Hershberger, H. D. and Hershberger, R. (1982). *Kuku-Yalanji dictionary*, volume 7 of *Work Papers of SIL - AAB. Series B*. Summer Institute of Linguistics, Darwin.

Hirsimäki, T., Creutz, M., Siivola, V., and Kurimo, M. (2003). Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proceedings of Eurospeech 2003, Geneva*, pages 2293–2996. Geneva, Switzerland.

Hirsimäki, T., Creutz, M., Siivola, V., and Kurimo, M. (2005). Morphologically motivated language models in speech recognition. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR '05), 15-17 June, Espoo, Finland*, pages 121–126. Espoo.

Hirsimäki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S., and Pylkknen, J. (to appear). Unlimited vocabulary speech recognition with morph language models applied to finnish. *Computer Speech and Language*.

Hu, Y., Matveeva, I., Goldsmith, J., and Sprague, C. (2005a). Refining the SED heuristic for morpheme discovery: Another look at Swahili. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 28–35, Ann Arbor, Michigan. Association for Computational Linguistics.

Hu, Y., Matveeva, I., Goldsmith, J., and Sprague, C. (2005b). Using morphology and syntax together in unsupervised learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 20–27, Ann Arbor, Michigan. Association for Computational Linguistics.

Hughes, B., Baldwin, T., Bird, S., Nicholson, J., and MacKinlay, A. (2006). Reconsidering language identification for written language resources. In *Proceedings 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 485–488. Genoa, Italy.

Hull, D. A. (1996). Stemming algorithms: A case study for detailed evaluation. *Journal of the American Soicety for Information Science*, 47(1):70–84.

Jacquemin, C. (1997). Guessing morphology from terms and corpora. In *Proceedings, 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97), Philadelphia, PA*, pages 165–165.

Janßen, A. (1992). Segmentierung französischer wortformen ohne lexikon. In Klenk, U., editor, *Computatio Linguae: Aufsätze zur algorithmischen und Quantitativen Analyse der Sprache*, volume 73 of *Zeitschrift für Dialektologie und Linguistik: Beihefte*, pages 74–95. Franz Steiner, Stuttgart.

Johnson, H. and Martin, J. (2003). Unsupervised learning of morphology for English and Inuktitut. In *HLT-NAACL 2003, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, May 27 - June 1, Edmonton, Canada*, volume Companion Volume - Short papers.

Julien, M. (2006). Word. In Brown, K., editor, *Encyclopedia of Language and Linguistics*, volume 13, pages 617–624. Elsevier, Amsterdam, 2 edition.

Juola, P. (2006). Language identification, automatic. In Brown, K., editor, *Encyclopedia of Language and Linguistics*, volume 6, pages 508–510. Elsevier, Amsterdam, 2 edition.

Karagol-Ayan, B., Doermann, D., and Weinberg, A. (2006). Morphology induction from limited noisy data using approximate string matching. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on*

Computational Phonology and Morphology at HLT-NAACL 2006, pages 60–68. Association for Computational Linguistics, New York City, USA.

Kazakov, D. (1997). Unsupervised learning of naïve morphology with genetic algorithms. In Daelemans, W., Weijters, T., and van der Bosch, A., editors, *ECML'97 – Workshop Notes on Empirical Learning of Natural Language Tasks*, pages 105–112, Prague. University of Economics.

Kazakov, D. (2000). Achievements and prospects of learning word morphology with inductive logic programming. In Cussens, J. and Dzeroski, S., editors, *Learning Language in Logic*, volume 1925 of *Lecture Notes in Computer Science*, pages 89–109. Springer-Verlag, Berlin.

Kazakov, D. and Manandhar, S. (1998). A hybrid approach to word segmentation. In Page, C. D., editor, *Proceedings of the 8th International Workshop on Inductive Logic Programming (ILP-98) in Madison, Wisconsin, USA*, volume 1446 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin.

Kazakov, D. and Manandhar, S. (2001). Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. *Machine Learning*, 43:121–162.

King James (1977). *The Holy Bible, containing the Old and New Testaments and the Apocrypha in the authorized King James version*. Thomas Nelson, Nashville, New York.

Klenk, U. (1985a). Ein nicht-lexikalisches verfahren zur erkennung spanischer wortstämme. In Klenk, U., editor, *Strukturen und Verfahren in der maschinellen Sprachverarbeitung*, pages 47–65. AQ-Verlag, Dudweiler. Not seen.

Klenk, U. (1985b). Recognition of spanish inflectional endings based on the distribution of characters. In Hamesse, J. and Zampolli, A., editors, *Computers in literary and linguistic computing: proceedings of the eleventh International Conference / L'ordinateur et les recherches littéraires et linguistiques: actes de la XIe Conférence internationale, Université catholique de Louvain (Louvain-la-Neuve) 2-6 avril 1984*, volume 30 of *Travaux de linguistique quantitative*, pages 246–253.

Klenk, U. (1991). Verfahren der segmentierung von wörtern in morphe: Mit einer untersuchung zum spanischen. In und Dieter Seelbach, J. R., editor, *Romanistische Computerlinguistik: Theorien und Implementationen*, volume 266 of *Linguistische Arbeiten*, pages 197–206. Niemeyer, Tübingen.

Klenk, U. (1992). Verfahren morphologischer segmentierung und die wortstruktur des spanischen. In Klenk, U., editor, *Computatio Linguae:*

Aufsätze zur algorithmischen und Quantitativen Analyse der Sprache, volume 73 of *Zeitschrift für Dialektologie und Linguistik: Beihefte*, pages 110–124. Franz Steiner, Stuttgart.

Klenk, U. (1994). Automatische morphologische analyse arabischer wortformen. In Klenk, U., editor, *Computatio Linguae II: Aufsätze zur algorithmischen und Quantitativen Analyse der Sprache*, volume 83 of *Zeitschrift für Dialektologie und Linguistik: Beihefte*, pages 84–101. Franz Steiner, Stuttgart.

Klenk, U. and Langer, H. (1989). Morphological segmentation without a lexicon. *Literary and Linguistic Computing*, 4(4):247–253.

Kontorovich, L., Don, D., and Singer, Y. (2003). A markov model for the acquisition of morphological structure. Technical report, CMU-CS-03-147, School of Computer Science, Carnegie Mellon University.

Kruengkrai, C., Srichaivattana, P. andSornlertlamvanich, V., and Isahara, H. (2005). Language identification based on string kernels. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005*, volume 2, pages 926–929.

Kurimo, M., Creutz, M., Varjokallio, M., Arisoy, E., and Saraclar, M. (2005). An introduction and evaluation report. In Kurimo, M., Creutz, M., and Lagus, K., editors, *Unsupervised segmentation of words into morphemes – Challenge 2005*.

Ladefoged, P. (2005). *Vowels and Consonants*. Blackwell, Oxford, 2 edition.

Langer, H. (1991). *Ein automatisches Morphsegmentierungsverfahren für deutsche Wortformen*. PhD thesis, Georg-August-Universität zu Göttingen.

Lefebvre, C. (2004). *Issues in the study of Pidgin and Creole languages*, volume 70 of *Studies in Language Companion Series*. John Benjamins, Amsterdam.

Lehmann, H. (1973). *Linguistische Modellbildung und Methodologie*. Max Niemeyer Verlag, Tübingen. Pp. 71-76 and 88-93.

Leizarraga, J. (1571). *Iesus Krist Gure Iaunaren Testamentu Berria*. Pierre Hautin, Inprimizale, Roxellan. NT only.

Lepage, Y. (1998). Solving analogies on words: an algorithm. In *Proceedings of the 17th international conference on Computational linguistics*, pages 728–734, Morristown, NJ, USA. Association for Computational Linguistics.

Lins, R. D. and Gonçalves, Jr., P. (2004). Automatic language identification of written texts. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1128–1133, New York, NY, USA. ACM Press.

Manandhar, S., Dzeroski, S., and Erjavec, T. (1998). Learning multilingual morphology with clog. In *Proceedings of the 8th International Workshop on Inductive Logic Programming*, pages 135–144. Springer-Verlag.

Manning, C. D. (1998). The segmentation problem in morphology learning. In Burstein, J. and Leacock, C., editors, *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Language Learning*, pages 299–305. Association for Computational Linguistics, Somerset, New Jersey.

Martin, T., Baker, B., Wong, E., and Sridharan, S. (2006). A syllable-scale framework for language identification. *Computer Speech & Language*, 20(2-3):276–302.

Martins, B. and Silva, M. J. (2005). Language identification in web pages. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 764–768, New York, NY, USA. ACM Press.

McNamee, P. (2005). Language identification: a solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101.

Medina Urrea, A. (2000). Automatic discovery of affixes by means of a corpus: A catalog of spanish affixes. *Journal of Quantitative Linguistics*, 7(2):97–114.

Medina Urrea, A. (2003). *Investigación cuantitativa de afijos y clíticos del español de México: Glutinometría en el* Corpus del Español Mexicano Contemporáneo. PhD thesis, El Colegio de México, México, D.F.

Medina-Urrea, A. (2006). Affix discovery by means of corpora: Experiments for spanish, czech, ralámuli and chuj. In Mehler, A. and Köhler, R., editors, *Aspects of Automatic Text Analysis*, volume 209 of *Studies in Fuzziness and Soft Computing*, pages 277–299. Springer, Berlin.

Medina Urrea, A. (2006). Towards the automatic lemmatization of 16th century mexican spanish: A stemming scheme for the chem. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing, 7th International Conference, CICLing 2006, Mexico City, Mexico, February 19-25, 2006, Proceedings*, volume 3878 of *Lecture Notes in Computer Science*, pages 101–104. Springer-Verlag, Berlin.

Medina Urrea, A. and Díaz, E. C. B. (2003). Características cuantitativas de la flexión verbal del chuj. *Estudios de Lingüística Aplicada*, 38:15–31.

Melucci, M. and Orio, N. (2003). A novel method for stemmer generation based on hidden markov models. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 131–138, New York, NY, USA. ACM Press.

Monson, C. (2004). A framework for unsupervised natural language morphology induction. In van der Beek, L. and andDaniel Midgley, D. G., editors, *ACL 2004: Student Research Workshop*, pages 67–72, Barcelona, Spain. Association for Computational Linguistics.

Monson, C., Carbonell, J., Lavie, A., and Levin, L. (2007). Paramor: Minimally supervised induction of paradigm structure and morphological analysis. In *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, pages 117–125, Prague, Czech Republic. Association for Computational Linguistics.

Monson, C., Lavie, A., Carbonell, J., and Levin, L. (2004). Unsupervised induction of natural language morphology inflection classes. In *SIGPHON 2004: Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pages 52–61, Barcelona, Spain. Association for Computational Linguistics.

Murthy, K. N. and Kumar, G. B. (2006). Language identification from small text samples. *Journal of Quantitative Linguistics*, 13(1):57–80.

Muthusamy, Y. K. and Spitz, L. A. (1997). Automatic language identification. In Cole, R. A., editor, *Survey of the State of the Art in Human Language Technology*, chapter 8.7. Center for Spoken Language Understanding CSLU, Carnegie Mellon University, Pittsburgh, PA.

Nash, D. G. (1980). *Topics in Warlpiri Grammar*. PhD thesis, Massachusetts Institute of Technology. Also published by Garland 1986.

Neuvel, S. and Fulop, S. A. (2002). Unsupervised learning of morphology without morphemes. In *Workshop on Morphological and Phonological Learning at Association for Computational Linguistics 40th Anniversary Meeting (ACL-02), July 6-12*, pages 9–15. ACL Publications.

Nunzio, G. D., Ferro, N., Melucci, M., and Orio, N. (2004). Experiments to evaluate probabilistic models for automatic stemmer generation and query word translation. In Peters, C., Braschler, M., Gonzalo, J., and Kluck, M., editors, *Proceedings of the Cross-Language Evaluation Forum (CLEF): Methodology and Metrics (CLEF 2003)*, volume 3237 of *Lecture Notes in Computer Science*, pages 220–235. Springer-Verlag, Berlin.

Oliver, A. (2004). *Adquisició d'informació lèxica i morfosintàctica a partir de corpus sense anotar: aplicació al rus i al croat*. PhD thesis, Universitat de Barcelona.

Patz, E. (2002). *A Grammar of the Kuku Yalanji Language of North Queensland*, volume 257 of *Pacific Linguistics*. Research School of Pacific and Asian Studies, Australian National University, Canberra.

Petzell, M. (2007). *A lingustic description of Kagulu*. PhD thesis, Göteborgs Universitet.

Pirkola, A. (2001). Morphological typology of languages for IR. *Journal of Documentation*, 57(3):330–348.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Poutsma, A. (2002). Applying monte carlo techniques to language identification. In Mariët, T., Nijholt, A., and Hondorp, H., editors, *Computational Linguistics in the Netherlands 2001: Selected Papers from the Twelfth CLIN Meeting*, volume 45 of *Language and Computers - Studies in Practical Linguistics*, pages 179–189. Rodopi, Amsterdam/New York, NY.

Prager, J. M. (2000). Linguini: Language identification for multilingual documents. *Journal of Management Information Systems*, 16(3):71–102.

Rai, N. K. (1985). *A descriptive study of Bantawa*. PhD thesis, Poona University.

Rogati, M., McCarley, S., and Yang, Y. (2003). Unsupervised learning of arabic stemming using a parallel corpus. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 391–398, Morristown, NJ, USA. Association for Computational Linguistics.

Rosenthal, F. (1995). *A grammar of biblical Aramaic*, volume 5 of *Porta linguarum Orientalium*. Harrassowitz, Wiesbaden, 6 edition.

Russell, K. (1999). The "word" in two polysynthetic languages. In Hall, A. T. and Kleinheiz, U., editors, *Studies on the Phonological Word*, volume 174 of *Current Issues in Linguistic Theory*, pages 203–221. John Benjamins, Amsterdam.

Sanders, G. (1990). On the analysis and implications of maori verb alternations. *Lingua*, 80:149–196.

Schone, P. (2001). *Toward Knowledge-Free Induction of Machine-Readable Dictionaries*. PhD thesis, University of Colorado.

Schone, P. and Jurafsky, D. (2000). Knowledge-free induction of inflectional morphologies using latent semantic analysis. In *Conference on Natural Language Learning 2000 (CoNLL-2000), Lisbon, Portugal*.

Schone, P. and Jurafsky, D. (2001). Knowledge-free induction of inflectional morphologies. In *Proceedings of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, PA, 2001.*

Sharma, U., Kalita, J., and Das, R. (2002). Unsupervised learning of morphology for building lexicon for a highly inflectional language. In *Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON), Philadelphia, July 2002*, pages 1–10. Association for Computational Linguistics.

Sibun, P. and Reynar, J. C. (1996). Language identification: Examining the issues. In *5th Symposium on Document Analysis and Information Retrieval*, pages 125–135, Las Vegas, Nevada, U.S.A.

Snover, M. G. (2002). An unsupervised knowledge free algorithm for the learning of morphology in natural languages. Master's thesis, Department of Computer Science, Washington University.

Snover, M. G. and Brent, M. R. (2001). A bayesian model for morpheme and paradigm identification. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 482–490. Morgan Kaufmann Publishers.

Snover, M. G. and Brent, M. R. (2003). A probabilistic model for learning concatenative morphology. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 1513–1520. MIT Press, Cambridge, MA.

Snover, M. G., Jarosz, G. E., and Brent, M. R. (2002). Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In *Workshop on Morphological and Phonological Learning at Association for Computational Linguistics 40th Anniversary Meeting (ACL-02), July 6-12*. ACL Publications.

Summer Institute of Linguistics (1985). Bible: New testament and old testament selctions in kuku-yalanji.

Summer Institute of Linguistics (2001). *Bible: selections in Warlpiri*. Document 0650 of the Aboriginal Studies Electronic Data Archive (ASEDA), AIATSIS (Australian Institute of Aboriginal and Torres Strait Islander Studies), Canberra. Translated in portions 1968–2001.

Svenska Bibelsällskapet (1917). *Gamla och Nya testamentet: de kanoniska böckerna*. Norstedt, Stockgholm.

Swan, O. E. (2002). *A grammar of contemporary Polish*. Slavica, Bloomington.

Takci, H. and Sogukpinar, I. (2004). Centroid-based language identification using letter feature set. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing: 5th International Conference, CICLing 2004 Seoul, Korea, February 15-21, 2004 Proceedings*, volume 2945 of *Lecture Notes in Computer Science*, pages 640–648. Springer-Verlag, Berlin.

The British & Foreign Bible Society (1996). *Maori Bible*. The British & Foreign Bible Society, London, England.

Thurmair, G. (1986). Ein morphologisches prozessorsegment zur erzeugung von grundformen mithilfe von lernverfahren. In Schwarz, C. and Thurmair, G., editors, *Informationslinguistische Texterschließung*, volume 4 of *Linguistische Datenverarbeitung*, pages 8–31. Georg Olms, Hildesheim.

Traill, A. (1994). *A !Xóõ Dictionary*, volume 9 of *Quellen zur Khoisan-Forschung/Research in Khoisan Studies*. Rüdiger Köppe Verlag, Köln.

Underhill, R. (1976). *Turkish Grammar*. MIT Press, Cambridge, MA.

Wicentowski, R. (2002). *Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. PhD thesis, Johns Hopkins University, Baltimore, MD.

Wicentowski, R. (2004). Multilingual noise-robust supervised morphological analysis using the wordframe model. In *Proceedings of the ACL Special Interest Group on Computational Phonology (SIGPHON)*, pages 70–77.

Williams, H. W. (1971). *A dictionary of the Maori language*. GP Books, Wellington, 7 edition.

Wothke, K. (1984). *Maschinelle Erlernung und Simulation morphologischer Ableitungs regeln*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität, Bonn.

Wothke, K. and Schmidt, R. (1992). A morphological segmentation procedure for german. *Sprache und Datenverarbeitung*, 16(1):15–28.

Xafopoulos, A., Kotropoulos, C., Almpanidis, G., and Pitas, I. (2004). Language identification in web documents using discrete HMMs. *Pattern Recognition*, 37(3):583–594(12).

Xanthos, A. (2007). *Apprentissage automatique de la morphologie: Le cas des structures racine-schème*. PhD thesis, Université de Lausanne.

Xanthos, A., Hu, Y., and Goldsmith, J. (2006). Exploring variant definitions of pointer length in mdl. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology at HLT-NAACL 2006*, pages 32–40. Association for Computational Linguistics, New York City, USA.

Yarowsky, D. and Wicentowski, R. (2000). Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 207–216.

Ziegler, D.-V. (1991). *The Automatic Identification of Languages Using Linguistic Recognition Signals*. PhD thesis, University of New York at Buffalo.

Zweigenbaum, P., Hadouche, F., and Grabar, N. (2003). Apprentissage de relations morphologiques en corpus. In Daille, B., editor, *Actes de TALN 2003*, pages 285–294. Batz-sur-mer, France.