Language Model Based Arabic Word Segmentation

Young-Suk Lee Kishore Papineni Salim Roukos
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Ossama Emam Hany Hassan
IBM Cairo Technology Development Center
P.O.Box 166, El-Ahram, Giza, Egypt

Abstract

We approximate Arabic's rich morphology by a model that a word consists of a sequence of morphemes in the pattern *prefix*-stem-suffix** (* denotes zero or more occurrences of a morpheme). Our method is seeded by a small manually segmented Arabic corpus and uses it to bootstrap an unsupervised algorithm to build the Arabic word segmenter from a large unsegmented Arabic corpus. The algorithm uses a trigram language model to determine the most probable morpheme sequence for a given input. The language model is initially estimated from a small manually segmented corpus of about 110,000 words. To improve the segmentation accuracy, we use an unsupervised algorithm for automatically acquiring new stems from a 155 million word unsegmented corpus, and re-estimate the model parameters with the expanded vocabulary and training corpus. The resulting Arabic word segmentation system achieves around 97% exact match accuracy on a test corpus containing 28,449 word tokens. We believe this is a state-of-the-art performance and the algorithm can be used for many highly inflected languages provided that one can create a small manually segmented corpus of the language of interest.

1 Introduction

Morphologically rich languages like Arabic present significant challenges to many natural language processing applications because a word often conveys complex meanings decomposable into several morphemes (i.e. prefix, stem, suffix). By segmenting words into morphemes, we can improve the performance of natural language systems including machine translation (Brown et al. 1993) and information retrieval (Franz, M. and McCarley, S. 2002). In this paper, we present a general word segmentation algorithm for handling inflectional morphology capable of segmenting a word into a *prefix*-stem-suffix** sequence, using a small manually segmented corpus and a table of prefixes/suffixes of the language. We do not address Arabic infix morphology where many stems correspond to the same root with various infix variations; we treat all the stems of a common root as separate atomic units. The use of a stem as a morpheme (unit of meaning) is better suited than the use of a root for the applications we are considering in information retrieval and machine translation (e.g. different stems of the same root translate into different English words.) Examples of Arabic words and their segmentation into *prefix*-stem-suffix** are given in Table 1, where '#' indicates a morpheme being a prefix, and '+' a suffix.[1] As

[1] Arabic is presented in both native and Buckwalter transliterated Arabic whenever possible. All native Arabic is to be read from right-to-left, and transliterated Arabic is to be read from left-to-right. The convention of

shown in Table 1, a word may include multiple prefixes, as in للـ (*l: for, Al*: the), or multiple suffixes, as in تـه (*t:* feminine singular, *h*: his). A word may also consist only of a stem, as in الـى (*AlY*, to/towards).

The algorithm implementation involves (i) language model training on a morpheme-segmented corpus, (ii) segmentation of input text into a sequence of morphemes using the language model parameters, and (iii) unsupervised acquisition of new stems from a large unsegmented corpus. The only linguistic resources required include a small manually segmented corpus ranging from 20,000 words to 100,000 words, a table of prefixes and suffixes of the language and a large unsegmented corpus.

In Section 2, we discuss related work. In Section 3, we describe the segmentation algorithm. In Section 4, we discuss the unsupervised algorithm for new stem acquisition. In Section 5, we present experimental results. In Section 6, we summarize the paper.

2 Related Work

Our work adopts major components of the algorithm from (Luo & Roukos 1996): language model (LM) parameter estimation from a segmented corpus and input segmentation on the basis of LM probabilities. However, our work diverges from their work in two crucial respects: (i) new technique of computing all possible segmentations of a word into *prefix*-stem-suffix** for decoding, and (ii) unsupervised algorithm for new stem acquisition based on a stem candidate's similarity to stems occurring in the training corpus.

(Darwish 2002) presents a supervised technique which identifies the root of an Arabic word by stripping away the prefix and the suffix of the word on the basis of manually acquired dictionary of word-root pairs and the likelihood that a prefix and a suffix would occur with the template from which the root is derived. He reports 92.7% segmentation accuracy on a 9,606 word evaluation corpus. His technique pre-supposes at most one prefix and one suffix per stem regardless of the actual number and meanings of prefixes/suffixes associated with the stem. (Beesley 1996) presents a finite-state morphological analyzer for Arabic, which displays the root, pattern, and prefixes/suffixes. The analyses are based on manually acquired lexicons and rules. Although his analyzer is comprehensive in the types of knowledge it presents, it has been criticized for their extensive development time and lack of robustness, cf. (Darwish 2002).

(Yarowsky and Wicentowsky 2000) presents a minimally supervised morphological analysis with a performance of over 99.2% accuracy for the 3,888 past-tense test cases in English. The core algorithm lies in the estimation of a probabilistic alignment between inflected forms and root forms. The probability estimation is based on the lemma alignment by frequency ratio similarity among different inflectional forms derived from the same lemma, given a table of inflectional parts-of-speech, a list of the canonical suffixes for each part of speech, and a list of the candidate noun, verb and adjective roots of the language. Their algorithm does not handle multiple affixes per word.

(Goldsmith 2000) presents an unsupervised technique based on the expectation-maximization algorithm and minimum description length to segment exactly one suffix per word, resulting in an F-score of 81.8 for suffix identification in English according to (Schone and Jurafsky 2001). (Schone and Jurafsky 2001) proposes an unsupervised algorithm capable of automatically inducing the morphology of inflectional languages using only text corpora. Their algorithm combines cues from orthography, semantics, and contextual information to induce morphological relationships in German, Dutch, and English, among others. They report F-scores between 85 and 93 for suffix analyses and between 78 and 85 for circumfix analyses in these languages. Although their algorithm captures prefix-suffix combinations or circumfixes, it does not handle the multiple affixes per word we observe in Arabic.

marking a prefix with '#" and a suffix with '+' will be adopted throughout the paper.

Words		Prefixes		Stems		Suffixes	
Arabic	Translit.	Arabic	Translit.	Arabic	Translit.	Arabic	Translit.
الولايات	*AlwlAyAt*	ال#	*Al#*	ولاي	*wlAy*	+ ات	*+At*
حياته	*HyAth*			حيا	*HyA*	+ت +ه	*+t +h*
للحصول	*llHSwl*	ل# ال#	*l# Al#*	حصول	*HSwl*		
الى	*AlY*			الى	*AlY*		

Table 1 Segmentation of Arabic Words into Prefix*-Stem-Suffix*

3 Morpheme Segmentation

3.1 Trigram Language Model

Given an Arabic sentence, we use a trigram language model on morphemes to segment it into a sequence of morphemes $\{m_1, m_2, ..., m_n\}$. The input to the morpheme segmenter is a sequence of Arabic tokens – we use a tokenizer that looks only at white space and other punctuation, e.g. quotation marks, parentheses, period, comma, etc. A sample of a manually segmented corpus is given below[2]. Here multiple occurrences of prefixes and suffixes per word are marked with an underline.

و# كان ايرفاين الذي حل في ال# مركز ال# اول في جائز +ة ال# نمسا ال# عام ال# ماضي علي سيار +ة فيراري شعر ب# الام في بطن +ه اضطر +ت +ه الي ال# انسحاب من ال# تجارب و# هو س# ي# عود الي لندن ل# اجراء ال# فحوص +ات ال# ضروري +ة حسب ما اشار فريق جاغوار . و# س# ي# حل سائق ال# تجارب في جاغوار ال# برازيلي لوسيانو بورتي مكان ايرفاين في ال# سباق غدا ال# احد الذي س# ي# كون اولي خطو +ات +ه في عالم سباق +ات الفورمولا

w# kAn AyrfAyn Al*y Hl fy Al# mrkz Al# Awl fy jA}z +p Al# nmsA Al# EAm Al# mADy Ely syAr +p fyrAry $Er b# AlAm fy bTn +h ADTr **+t +h** Aly Al# AnsHAb mn Al# tjArb w# hw **s# y#** Ewd Aly lndn l# AjrA' Al# fHwS +At Al# Drwry +p Hsb mA A$Ar fryq jAgwAr. **w# s# y#** Hl sA}q Al# tjArb fy jAgwAr Al# brAzyly lwsyAnw bwrty mkAn AyrfAyn fy Al# sbAq gdA Al# AHd Al*y **s# y#** kwn Awly xTw **+At +h** fy EAlm sbAq +At Alfwrmwla

Many instances of prefixes and suffixes in Arabic are meaning bearing and correspond to a word in English such as pronouns and prepositions. Therefore, we choose a segmentation into multiple prefixes and suffixes. Segmentation into one prefix and one suffix per word, cf. (Darwish 2002), is not very useful for applications like statistical machine translation, (Brown et al. 1993), for which an accurate word-to-word alignment between the source and the target languages is critical for high quality translations.

The trigram language model probabilities of morpheme sequences, $p(m_i|m_{i-1}, m_{i-2})$, are estimated from the morpheme-segmented corpus. At token boundaries, the morphemes from previous tokens constitute the histories of the current morpheme in the trigram language model. The trigram model is smoothed using deleted interpolation with the bigram and unigram models, (Jelinek 1997), as in (1):

(1) $p(m_3 \mid m_1, m_2) = \lambda_3\, p(m_3 \mid m_1, m_2) + \lambda_2\, p(m_3 \mid m_2) + \lambda_3\, p(m_3)$, where $\lambda_1+\lambda_2+\lambda_3 = 1$.

A small morpheme-segmented corpus results in a relatively high out of vocabulary rate for the stems. We describe below an unsupervised acquisition of new stems from a large unsegmented Arabic corpus. However, we first describe the segmentation algorithm.

3.2 Decoder for Morpheme Segmentation

[2] A manually segmented Arabic corpus containing about 140K word tokens has been provided by LDC (http://www.ldc.upenn.edu). We divided this corpus into training and the development test sets as described in Section 5.

We take the unit of decoding to be a sentence that has been tokenized using white space and punctuation. The task of a decoder is to find the morpheme sequence which maximizes the trigram probability of the input sentence, as in (2):

(2) $\text{SEGMENTATION}_{best} = Argmax\ \Pi_{i=1,\ N}\ p(m_i|m_{i-1}m_{i-2})$, N = number of morphemes in the input.

Search algorithm for (2) is informally described for each word token as follows:

Step 1: Compute all possible segmentations of the token (to be elaborated in 3.2.1).
Step 2: Compute the trigram language model score of each segmentation. For some segmentations of a token, the stem may be an out of vocabulary item. In that case, we use an "UNKNOWN" class in the trigram language model with the model probability given by $p(UNKNOWN|m_{i-1}, m_{i-2})$ * *UNK_Fraction,* where *UNK_Fraction* is 1e-9 determined on empirical grounds. This allows us to segment new words with a high accuracy even with a relatively high number of unknown stems in the language model vocabulary, cf. experimental results in Tables 5 & 6.
Step 3: Keep the top N highest scored segmentations.

3.2.1 Possible Segmentations of a Word

Possible segmentations of a word token are restricted to those derivable from a table of prefixes and suffixes of the language for decoder speed-up and improved accuracy.

Table 2 shows examples of atomic (e.g. ال, ات) and multi-component (e.g. وبـال, اتهـا) prefixes and suffixes, along with their component morphemes in native Arabic.[3]

Prefixes		Suffixes	
ال	ال#	ات	+ات
بـال	ب# ال#	اتهـا	+ات +ها
وبـال	و# ب# ال#	ونـهم	+ون +هم

Table 2 Prefix/Suffix Table

Each token is assumed to have the structure *prefix*-stem-suffix**, and is compared against the prefix/suffix table for segmentation. Given a word token, (i) identify all of the matching prefixes and suffixes from the table, (ii) further segment each matching prefix/suffix at each character position, and (iii) enumerate all *prefix*-stem-suffix** sequences derivable from (i) and (ii).

Table 3 shows all of its possible segmentations of the token واكررها (*wAkrrhA*; 'and I repeat it'),[4] where ∅ indicates the null prefix/suffix and the *Seg Score* is the language model probabilities of each segmentation S1 ... S12. For this token, there are two matching prefixes و# (*w#*) and و ا# (*wA#*) from the prefix table, and two matching suffixes + ا (*+A*) and + هـا (*+hA*) from the suffix table. S1, S2, & S3 are the segmentations given the null prefix ∅ and suffixes ∅, *+A*, *+hA*. S4, S5, & S6 are the segmentations given the prefix *w#* and suffixes ∅, *+A, +hA*. S7, S8, & S9 are the segmentations given the prefix *wA#* and suffixes ∅, *+A, +hA*. S10, S11, & S12 are the segmentations given the prefix sequence *w# A#* derived from the prefix *wA#* and suffixes ∅, *+A, +hA*. As illustrated by S12, derivation of sub-segmentations of the matching prefixes/suffixes enables the system to identify possible segmentations which would have been missed otherwise. In this case, segmentation including the derived prefix sequence و# ا# كـرر +هـا (***w# A#*** *krr +hA*) happens to be the correct one.

3.2.2. Prefix-Suffix Filter

While the number of possible segmentations is maximized by sub-segmenting matching

[3] We have acquired the prefix/suffix table from a 110K word manually segmented LDC corpus (51 prefixes & 72 suffixes) and from IBM-Egypt (additional 14 prefixes & 122 suffixes). The performance improvement by the additional prefix/suffix list ranges from 0.07% to 0.54% according to the manually segmented training corpus size. The smaller the manually segmented corpus size is, the bigger the performance improvement by adding additional prefix/suffix list is.

[4] A sentence in which the token occurs is as follows: قلتها واكررها فالمشكلة ليست في النفط الخام وانما في المشتقات النفطية (*qlthA wAkrrhA fAlm$klp lyst fy AlfnT AlxAm wAnmA fy Alm$tqAt AlnfTyp.*)

prefixes and suffixes, some of illegitimate sub-segmentations are filtered out on the basis of the knowledge specific to the manually segmented corpus. For instance, sub-segmentation of the suffix *hA* into *+h +A* is ruled out because there is no suffix sequence *+h +A* in the training corpus. Likewise, sub-segmentation of the prefix *Al* into *A# l#* is filtered out. Filtering out improbable prefix/suffix sequences improves the segmentation accuracy, as shown in Table 5.

	Prefix	Stem	Suffix	Seg Scores
S1	∅	*wAkrrhA*	∅	2.6071e-05
S2	∅	*wAkrrh*	*+A*	1.36561e-06
S3	∅	*wAkrr*	*+hA*	9.45933e-07
S4	*w#*	*AkrrhA*	∅	2.72648e-06
S5	*w#*	*Akrrh*	*+A*	5.64843e-07
S6	*w#*	*Akrr*	*+hA*	4.52229e-05
S7	*wA#*	*krrhA*	∅	7.58256e-10
S8	*wA#*	*krrh*	*+A*	5.09988e-11
S9	*wA#*	*krr*	*+hA*	1.91774e-08
S10	*w# A#*	*krrhA*	∅	7.69038e-07
S11	*w# A#*	*krrh*	*+A*	1.82663e-07
S12	***w# A#***	***krr***	***+hA***	**0.000944511**

Table 3 Possible Segmentations of واكررها (*wAkrrhA*)

4 Unsupervised Acquisition of New Stems

Once the seed segmenter is developed on the basis of a manually segmented corpus, the performance may be improved by iteratively expanding the stem vocabulary and retraining the language model on a large automatically segmented Arabic corpus.

Given a small manually segmented corpus and a large unsegmented corpus, segmenter development proceeds as follows.

Initialization: Develop the seed segmenter $Segmenter_0$ trained on the manually segmented corpus $Corpus_0$, using the language model vocabulary, $Vocab_0$, acquired from $Corpus_0$.
Iteration: For i = 1 to N, N = the number of partitions of the unsegmented corpus

i. Use $Segmenter_{i-1}$ to segment $Corpus_i$.

ii. Acquire new stems from the newly segmented $Corpus_i$. Add the new stems to $Vocab_{i-1}$, creating an expanded vocabulary $Vocab_i$.

iii. Develop $Segmenter_i$ trained on $Corpus_0$ through $Corpus_i$ with $Vocab_i$.

Optimal Performance Identification: Identify the $Corpus_i$ and $Vocab_i$, which result in the best performance, i.e. system training with $Corpus_{i+1}$ and $Vocab_{i+1}$ does not improve the performance any more.

Unsupervised acquisition of new stems from an automatically segmented new corpus is a three-step process: (i) select new stem candidates on the basis of a frequency threshold, (ii) filter out new stem candidates containing a sub-string with a high likelihood of being a prefix, suffix, or prefix-suffix. The likelihood of a sub-string being a prefix, suffix, and prefix-suffix of a token is computed as in (5) to (7), (iii) further filter out new stem candidates on the basis of contextual information, as in (8).

(5) P_{score} = number of tokens with prefix P / number of tokens starting with sub-string P
(6) S_{score} = number of tokens with suffix S / number of tokens ending with sub-string S
(7) PS_{score} = number of tokens with prefix P and suffix S / number of tokens starting with sub-string P and ending with sub-string S

Stem candidates containing a sub-string with a high prefix, suffix, or prefix-suffix likelihood are filtered out. Example sub-strings with the prefix, suffix, prefix-suffix likelihood 0.85 or higher in a 110K word manually segmented corpus are given in Table 4. If a token starts with the sub-string سنـ (*sn*), and end with ـها (*hA*), the sub-string's likelihood of being the prefix-suffix of the token is 1. If a token starts with the sub-string لل (*ll),* the sub-string's likelihood of being the prefix of the token is 0.945, etc.

Arabic	Transliteration	Score
سنـ # **stem** + ـها	*sn#* **stem**+*hA*	1.0
الـ# **stem** +ة	*Al#* **stem**+*p*	0.984
لل# **stem**	*ll#* **stem**	0.945
stem +ات	**stem**+*At*	0.889

Table 4 Prefix/Suffix Likelihood Score

(8) Contextual Filter: (i) Filter out stems co-occurring with prefixes/suffixes not present in the training corpus. (ii) Filter out stems whose prefix/suffix distributions are highly disproportionate to those seen in the training corpus.

According to (8), if a stem is followed by a potential suffix *+m*, not present in the training corpus, then it is filtered out as an illegitimate stem. In addition, if a stem is preceded by a prefix and/or followed by a suffix with a significantly higher proportion than that observed in the training corpus, it is filtered out. For instance, the probability for the suffix *+A* to follow a stem is less than 50% in the training corpus regardless of the stem properties, and therefore, if a candidate stem is followed by *+A* with the probability of over 70%, e.g. *mAnyl +A*, then it is filtered out as an illegitimate stem.

5 Performance Evaluations

We present experimental results illustrating the impact of three factors on segmentation error rate: (i) the base algorithm, i.e. language model training and decoding, (ii) language model vocabulary and training corpus size, and (iii) manually segmented training corpus size. Segmentation error rate is defined in (9).

(9) (*number of incorrectly segmented tokens / total number of tokens*) *x 100*

Evaluations have been performed on a development test corpus containing 28,449 word tokens. The test set is extracted from 20001115_AFP_ARB.0060.xml.txt through 20001115_AFP_ARB.0236.xml.txt of the LDC Arabic Treebank: Part 1 v 2.0 Corpus. Impact of the core algorithm and the unsupervised stem acquisition has been measured on segmenters developed from 4 different sizes of manually segmented seed corpora: 10K, 20K, 40K, and 110K words.

The experimental results are shown in Table 5. The baseline performances are obtained by assigning each token the most frequently occurring segmentation in the manually segmented training corpus. The column headed by '**3-gram LM**' indicates the impact of the segmenter using only trigram language model probabilities for decoding. Regardless of the manually segmented training corpus size, use of trigram language model probabilities reduces the word error rate of the corresponding baseline by approximately 50%. The column headed by '**3-gram LM + PS Filter**' indicates the impact of the core algorithm plus Prefix-Suffix Filter discussed in Section 3.2.2. Prefix-Suffix Filter reduces the word error rate ranging from 7.4% for the smallest (10K word) manually segmented corpus to 21.8% for the largest (110K word) manually segmented corpus —- around 1% absolute reduction for all segmenters. The column headed by '**3-gram LM + PS Filter + New Stems**' shows the impact of unsupervised stem acquisition from a 155 million word Arabic corpus. Word error rate reduction due to the unsupervised stem acquisition is 38% for the segmenter developed from the 10K word manually segmented corpus and 32% for the segmenter developed from 110K word manually segmented corpus.

Language model vocabulary size (LM VOC Size) and the unknown stem ratio (OOV ratio) of various segmenters is given in Table 6. For unsupervised stem acquisition, we have set the frequency threshold at 10 for every 10-15 million word corpus, i.e. any new morphemes occurring more than 10 times in a 10-15 million word corpus are considered to be new stem candidates. Prefix, suffix, prefix-suffix likelihood score to further filter out illegitimate stem candidates was set at 0.5 for the segmenters developed from 10K, 20K, and 40K manually segmented corpora, whereas it was set at 0.85 for the segmenters developed from a 110K manually segmented corpus. Both the frequency threshold and the optimal prefix, suffix, prefix-suffix likelihood scores were determined on empirical grounds. Contextual Filter stated in (8) has been applied only to the segmenter developed from 110K manually segmented training corpus.[5] Comparison of Tables 5 and 6 indicates a high correlation between the segmentation error rate and the unknown stem ratio.

[5] Without the Contextual Filter, the error rate of the same segmenter is 3.1%.

Manually Segmented Training Corpus Size	Baseline	3-gram LM	3-gram LM + PS Filter	3-gram LM + PS Filter + New Stems
10K Words	26.0%	14.7%	13.6%	**8.5%**
20K Words	19.7%	9.1%	8.0%	**5.9%**
40K Words	14.3%	7.6%	6.5%	**5.1%**
110K Words	11.0%	5.5%	4.3%	**2.9%**

Table 5 Impact of Core Algorithm and LM Vocabulary Size on Segmentation Error Rate

Manually Segmented Training Corpus Size	3-gram LM		3-gram LM + PS Filter + New Stems	
	LM VOC Size	OOV Ratio	LM VOC Size	OOV Ratio
10K Words	2,496	20.4%	22,964	7.8%
20K Words	4,111	11.4%	25,237	5.3%
40K Words	5,531	9.0%	21,156	4.7%
110K Words	8,196	5.8%	25,306	1.9%

Table 6 Language Model Vocabulary Size and Out of Vocabulary Ratio

Manually Segmented Training Corpus Size	3-gram LM + PS Filter + New Stems			
	Unknown Stem	*Alywm*	Other Errors	Total # of Errors
10 K Words	1,844 (76.9%)	98 (4.1%)	455 (19.0%)	2,397
20 K Words	1,174 (71.1%)	82 (5.0%)	395 (23.9%)	1,651
40 K Words	1,005 (69.9%)	81 (5.6%)	351 (24.4%)	1,437
110 K Words	333 (39.6%)	82 (9.8%)	426 (50.7%)	841

Table 7 Segmentation Error Analyses

Table 7 gives the error analyses of four segmenters according to three factors: (i) errors due to unknown stems, (ii) errors involving اليوم (*Alywm*), and (iii) errors due to other factors. Interestingly, the segmenter developed from a 110K manually segmented corpus has the lowest percentage of "unknown stem" errors at 39.6% indicating that our unsupervised acquisition of new stems is working well, as well as suggesting to use a larger unsegmented corpus for unsupervised stem acquisition.

اليوم (*Alywm*) should be segmented differently depending on its part-of-speech to capture the semantic ambiguities. If it is an adverb or a proper noun, it is segmented as اليوم 'today/Al-Youm', whereas if it is a noun, it is segmented as ال# يوم 'the day.' Proper segmentation of اليوم primarily requires its part-of-speech information, and cannot be easily handled by morpheme trigram models alone.

Other errors include over-segmentation of foreign words such as بوتين (*bwtyn*) as ب# وتين and ليتر (*lytr*) 'litre' as ل# ي# تر.

These errors are attributed to the segmentation ambiguities of these tokens: بوتين is ambiguous between 'بوتين (Putin)' and 'ب# وتين (by aorta)'. ليتر is ambiguous between 'ليتر (litre)' and 'ل# ي# تر (for him to harm)'. These errors may also be corrected by incorporating part-of-speech information for disambiguation.

To address the segmentation ambiguity problem, as illustrated by 'بوتين (Putin)' vs. 'ب# وتين (by aorta)', we have developed a joint model for segmentation and part-of-speech tagging for which the best segmentation of an input sentence is obtained according to the formula (10), where t_i is the part-of-speech of morpheme m_i, and N is the number of morphemes in the input sentence.

(10) $\text{SEGMENTATION}_{\text{best}} = \textit{Argmax} \prod_{i=1,N} p(m_i|m_{i-1}\, m_{i-2})\, p(t_i|t_{i-1}\, t_{i-2})\, p(m_i|t_i)$

By using the joint model, the segmentation word error rate of the best performing segmenter has been reduced by about 10%

from 2.9% (cf. the last column of Table 5) to 2.6%.

5 Summary and Future Work

We have presented a robust word segmentation algorithm which segments a word into a *prefix*-stem-suffix** sequence, along with experimental results. Our Arabic word segmentation system implementing the algorithm achieves around 97% segmentation accuracy on a development test corpus containing 28,449 word tokens. Since the algorithm can identify any number of prefixes and suffixes of a given token, it is generally applicable to various language families including agglutinative languages (Korean, Turkish, Finnish), highly inflected languages (Russian, Czech) as well as semitic languages (Arabic, Hebrew).

Our future work includes (i) application of the current technique to other highly inflected languages, (ii) application of the unsupervised stem acquisition technique on about 1 billion word unsegmented Arabic corpus, and (iii) adoption of a novel morphological analysis technique to handle irregular morphology, as realized in Arabic broken plurals كـتـاب (*ktAb*) 'book' vs. كـتـب (*ktb*) 'books'.

Acknowledgment

This work was partially supported by the Defense Advanced Research Projects Agency and monitored by SPAWAR under contract No. N66001-99-2-8916. The views and findings contained in this material are those of the authors and do not necessarily reflect the position of policy of the Government and no official endorsement should be inferred. We would like to thank Martin Franz for discussions on language model building, and his help with the use of *ViaVoice* language model toolkit.

References

Beesley, K. 1996. Arabic Finite-State Morphological Analysis and Generation. *Proceedings of COLING-96*, pages 89– 94.

Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. 1993. The mathematics of statistical machine translation: Parameter Estimation. *Computational Linguistics*, 19(2): 263–311.

Darwish, K. 2002. Building a Shallow Arabic Morphological Analyzer in One Day. *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 47–54.

Franz, M. and McCarley, S. 2002. Arabic Information Retrieval at IBM. *Proceedings of TREC 2002*, pages 402– 405.

Goldsmith, J. 2000. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(1).

Jelinek, F. 1997. *Statistical Methods for Speech Recognition*. The MIT Press.

Luo, X. and Roukos, S. 1996. An Iterative Algorithm to Build Chinese Language Models. *Proceedings of ACL-96*, pages 139–143.

Schone, P. and Jurafsky, D. 2001. Knowledge-Free Induction of Inflectional Morphologies. *Proceedings of North American Chapter of Association for Computational Linguistics*.

Yarowsky, D. and Wicentowski, R. 2000. Minimally supervised morphological analysis by multimodal alignment. *Proceedings of ACL-2000*, pages 207– 216.

Yarowsky, D, Ngai G. and Wicentowski, R. 2001. Inducting Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. *Proceedings of HLT 2001*, pages 161–168.