Latent Morpho-Semantic Analysis: Multilingual Information Retrieval with Character N-Grams and Mutual Information

Peter A. Chew, Brett W. Bader
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185, USA
{pchew,bwbader}@sandia.gov

Ahmed Abdelali
New Mexico State University
P.O. Box 30002, Mail Stop 3CRL
Las Cruces, NM 88003-8001, USA
ahmed@crl.nmsu.edu

Abstract

We describe an entirely statistics-based, unsupervised, and language-independent approach to multilingual information retrieval, which we call Latent Morpho-Semantic Analysis (LMSA). LMSA overcomes some of the shortcomings of related previous approaches such as Latent Semantic Analysis (LSA). LMSA has an important theoretical advantage over LSA: it combines well-known techniques in a novel way to break the terms of LSA down into units which correspond more closely to morphemes. Thus, it has a particular appeal for use with morphologically complex languages such as Arabic. We show through empirical results that the theoretical advantages of LMSA can translate into significant gains in precision in multilingual information retrieval tests. These gains are not matched either when a standard stemmer is used with LSA, or when terms are indiscriminately broken down into n-grams.

1 Introduction

As the linguistic diversity of textual resources increases, and need for access to those resources grows, there is also greater demand for efficient information retrieval (IR) methods which are truly language-independent. In the ideal but possibly unattainable case, an IR algorithm would produce equally reliable results for any language pair: for example, a query in English would retrieve equally good results in Arabic as in French.

A number of developments in recent years have brought that goal more within reach. One of the factors that severely hampered early attempts at machine translation, for example, was the lack of available computing power. However, Moore's Law, the driving force of change in computing since then, has opened the way for recent progress in the field, such as Statistical Machine Translation (SMT) (Koehn et al. 2003). Even more closely related to the topic of the present paper, implementations of the Singular Value Decomposition (SVD) (which is at the heart of LSA), and related algorithms such as PARAFAC2 (Harshman 1972), have become both more widely available and more powerful. SVD, for example, is available in both commercial off-the-shelf packages and at least one open-source implementation designed to run on a parallel cluster (Heroux et al. 2005).

Despite these advances, there are (as yet) not fully surmounted obstacles to working with certain language pairs, particularly when the languages are not closely related. This is demonstrated in Chew and Abdelali (2008). At least in part, this has to do with the lexical statistics of the languages concerned. For example, because Arabic has a much richer morphological structure than English and French (meaning is varied through the addition of prefixes and suffixes rather than separate terms such as particles), it has a considerably higher type-to-token

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 129–136
Manchester, August 2008

ratio. Exactly this type of language-specific statistical variation seems to lead to difficulties for statistics-based techniques such as LSA, as evidenced by lower cross-language information retrieval (CLIR) precision for Arabic/English than for French/English (Chew and Abdelali 2008).

In this paper, we present a strategy for overcoming these difficulties. In section 2, we outline the basic problem and the thinking behind our approach: that breaking words down into morphemes, or at least morphologically significant subconstituents, should enable greater inter-language comparability. This in turn should in theory lead to improved CLIR results. Several alternatives for achieving this are considered in section 3. One of these, a novel combination of mutual-information-based morphological tokenization (a step beyond simple n-gram tokenization) and SVD, is what we call LMSA. Section 4 discusses the framework for testing our intuitions, and the results of these tests are presented and discussed in section 5. Finally, we draw some conclusions and outline possible directions for future research in section 6.

2 The problem

In many approaches to IR, the underlying method is to represent a corpus as a term-by-document matrix in which each row corresponds to a unique term[2], and each column to a document in the corpus. The standard LSA framework (Deerwester et al. 1990) is no different, except that the (sparse) term-by-document matrix X is subjected to SVD,

$$X = USV^T \quad (1)$$

where U is a smaller but dense term-by-concept matrix, S is a diagonal matrix of singular values, and V is a dense document-by-concept matrix for the documents used in training. Effectively, U and V respectively map the terms and documents to a single set of arbitrary concepts, such that semantically related terms or documents (as determined by patterns of co-occurrence) will tend to be similar; similarity is usually measured by taking the cosine between two (term or document) vectors. New documents can also be projected into the LSA 'semantic space' by multiplying their document vectors (formed in exactly the same way as the columns for X) by the product US^{-1}, to yield document-by-concept vectors. LSA is a completely unsupervised approach to IR in that associations between terms simply fall out when SVD is applied to the data.

With cross-language or multilingual LSA, the approach differs little from that just outlined. The only required modification is in the training data: the term-by-document matrix must be formed from a parallel corpus, in which each document is the *combination* of text from the parallel languages (as described in Berry et al. 1994). Clearly, this IR model cannot be deployed to any languages not in the parallel corpus used for training SVD. However, recent work (Chew et al. 2007) shows not only that there is no limit (at least up to a certain point) to the number of languages that can be processed in parallel, but that precision actually increases for given language pairs as more other languages are included. In practice, the factors which limit the addition of parallel languages are likely to be computational power and the availability of parallel aligned text. As noted in section 1, the first of these is less and less of an issue; and regarding the second, parallel corpora (which are the mainstay of many current approaches to computational linguistics and IR, particularly in real-world applications) are becoming increasingly available. Substantially all of the Bible, in particular, is already electronically available in at least 40-50 languages from diverse language families (Biola University 2005-2006).

Yet, there are clearly variations in how well CLIR works. In previous results (Chew et al. 2007, Chew and Abdelali 2008) it is noticeable in particular that the results for Arabic and Russian (the two most morphologically complex languages for which they present results) are consistently poorer than they are for other languages. To our knowledge, no solution for this has been proposed and validated. Ideally, a solution would both make sense theoretically (or linguistically) and be statistics-based rather than rule-based, consistent with the general framework of LSA and other recent developments in the field, such as SMT, and avoiding the need to build a separate grammar for every new language – an expensive undertaking.

Translation	Types	Tokens	Ratio
English (KJV)	12,335	789,744	1.56%
French (Darby)	20,428	812,947	2.51%
Spanish (RV 1909)	28,456	704,004	4.04%
Russian (Syn 1876)	47,226	560,524	8.43%
Arabic (S. Van Dyke)	55,300	440,435	12.56%

Table 1. Lexical statistics in a parallel corpus

[2] Pragmatically, terms can be defined very straightforwardly in the regular expressions language as sequences of characters delimited by non-word characters.

To begin to assess the problem, one can compare the lexical statistics for the Bible from Chew et al. (2007), which should be directly comparable since they are from a parallel corpus. These are arranged in Table 1 in order of type-to-token ratio.

This ordering also corresponds to the ordering of languages on a scale from 'analytic' to 'synthetic': meaning is shaped in the former by the use of particles and word order, and in the latter by inflection and suffixation. Some examples illustrating differences between Russian and English in this respect are given in Table 2.

English	I read	you read	they read
Russian	читаю	читаешь	читают

Table 2. Analytic versus synthetic languages

The element in Russian, of course, which corresponds to 'read' is the stem 'чита', but this is embedded within a larger term. Hence, in all three examples, Russian takes one term to express what in English takes two terms. The same occurs (although to a lesser extent) in English, in which 'read' and 'reads' are treated as distinct terms. Without any further context (such as sentences in which these terms are instantiated), the similarity in meaning between 'read' and 'reads' will be readily apparent to any linguist, simply because of the shared orthography and morphology. But for an approach like standard LSA in which terms are defined simply as distinct entities delimited by non-word characters, the morphology is considered immaterial – it is invisible. The only way a standard term-based approach can detect any similarity between 'read' and 'reads' is through the associations of terms in documents. Clearly, then, such an approach operates under a handicap.

Two unfortunate consequences will inevitably result from this. First, some terms will be treated as out-of-vocabulary even when at least some of the semantics could perhaps have been derived from a part of the term. For example, if the training corpus contains 'read' and 'reads' but not 'reading', valuable information is lost every time 'reading' is encountered in a new document to which LSA might be deployed. Secondly, associations that should be made between *in-vocabulary* terms will also be missed. Perhaps a reason that more attention has not been devoted to this is that the problem can largely be disregarded in highly analytic languages like English. But, as previous results such as Chew and Abdelali's (2008) show, for a language like Arabic, the adverse consequences of a morphology-blind approach are more severe. The question then is: how can information which is clearly available in the training corpus be more fully leveraged without sacrificing efficiency?

3 Possible solutions

3.1 Replacing terms with n-grams

At first glance, one might think that stemming would be an answer. Stemming has been shown to improve IR, in particular for morphologically complex languages (recent examples, including with Arabic, are Lavie et al. 2004 and Abdou et al. 2005). We are not aware, however, of any previous results that show unequivocally that stemming is beneficial specifically in CLIR. Chew and Abdelali (2008) examine the use of a light stemmer for Arabic (Darwish 2002), and while this does result in a small overall increase in overall precision, there is paradoxically no increase for Arabic. The problem may be that the approach for Arabic needs to be matched by a similar approach for other languages in the parallel corpus. However, since stemmers are usually tailored to particular languages – and may even be unavailable for some languages – use of existing stemmers may not always be an option.

Another more obviously language-independent approach is to replace terms with character n-grams[3]. This is feasible for more or less any language, regardless of script. Moreover, implementation of a similar idea is described in McNamee and Mayfield (2004) and applied specifically to CLIR. However, McNamee and Mayfield's CLIR results are solely for European languages written in the Roman script. This is why they are able to obtain, in their words, 'surprisingly good results… without translation [of the query]', and without using LSA in any form. With related languages in the same script, and particularly when n-grams are used in place of terms, the existence of cognates means that many translations can easily be identified, since they probably share many of the same n-grams (e.g. French 'parisien' versus English 'Parisian'). When languages do not all share the same script or come from the same language family, however, the task can be considerably harder.

Since the approach of n-gram tokenization has the advantages of being entirely statistically-

[3] Hereafter, we use the term 'n-grams' to refer specifically to character (not word) n-grams.

based and language-independent, however, we examined whether it could be combined with LSA to allow CLIR (including cross-script retrieval), and whether this would lead to any advantage over term-based LSA. Our intuition was that some (although not all) n-grams would correspond to morphologically significant subconstituents of terms, such as 'read' from 'reading', and therefore associations at the morpheme level might be facilitated. The steps for this approach are listed in Table 3.

1	Form a term-by-document array from the parallel corpus as described above
2	For each term, list all (overlapping) n-grams[4]
3	Replace terms in the term-by-document array with n-grams, to form an n-gram-by-document array
4	Subject the n-gram-by-document array to SVD to produce an n-gram-by-concept U matrix, singular values (the diagonal S matrix), and document-by-concept V matrix
5	Project new documents into the semantic space by multiplying their vectors by US^{-1}

Table 3. Steps for n-gram-based LSA

Under all approaches, we selected the same log-entropy term weighting scheme that we used for standard LSA. Thus, whether a term *t* stands for a wordform or an n-gram, its weighted frequency W in a particular document *k* is given by:

$$W = \log_2 (F + 1) \times (1 + H_t / \log_2 (N))^{\alpha} \quad (2)$$

where F is the raw frequency of *t* in *k*, H_t is the entropy of the term or n-gram across all documents, N is the number of documents in the corpus, and α is some arbitrary constant (a power to which the global weight is raised). We have found that an $\alpha > 1$ improves precision by changing the relative distribution of weighted frequencies. Common terms with high entropy become much less influential in the SVD.

It should be noted that step (2) in Table 3 is similar to McNamee and Mayfield's approach, except that we did not include word-spanning n-grams, owing to computational constraints. We also tried two variants of step (2), one in which all n-grams were of the same length (as per McNamee and Mayfield 2004), and one in which n-grams of different lengths were mixed. Under the second of these, the number of rows in both the term-by-document and U matrices is of course considerably larger. For example, Table 4 shows that the number of rows in the n-gram-by-document matrix for English (EN) under the first variant (with n = 6) is 19,801, while under the second (with $n \leq 6$) it is 58,907. Comparable statistics are given for Arabic (AR), Spanish (ES), French (FR) and Russian (RU).

n=	AR	EN	ES	FR	RU
1	35	27	41	41	47
2	939	516	728	708	827
3	11,127	4,267	5,563	5,067	7,808
4	40,835	13,927	19,686	15,948	30,702
5	53,671	20,369	35,526	25,253	54,647
6	39,822	19,801	42,408	28,274	65,308
Total	146,429	58,907	103,952	75,291	159,339

Table 4. Number of distinct n-grams by language and length, up to length 6, based on Bible text

3.2 Replacing terms with morphemes: LMSA

We also attempted a related approach with *non-overlapping* n-grams. This set of experiments was guided by the intuition that not all n-grams are morphologically significant. Before we discuss the details of this approach, consider the English example 'comingle'. Here, 'co' + 'mingle' are likely to be more significant to the overall meaning than 'coming' + 'le' – in fact, the presence of the n-gram 'coming' could be misleading in this case. One way to model this would be to change the weighting scheme. The problem with this is that the weighting for one token has to be contingent on the weighting for another in the same term. Otherwise, in this example, the n-gram 'coming' would presumably receive a high weighting based on its frequency elsewhere in the corpus.

An alternative is to select the tokenization which maximizes mutual information (MI). Brown et al. (1992) describe one application of MI to identify word collocations; Kashioka et al. (1998) describe another, based on MI of character n-grams, for morphological analysis of Japanese. The pointwise MI of a pair s_1 and s_2 as adjacent symbols is

$$MI = \log P(s_1 s_2) - \log P(s_1) - \log P(s_2) \quad (3)$$

If s_1 follows s_2 less often than expected on the basis of their independent frequencies, then MI is negative; otherwise, it is positive.

In our application, we want to consider all candidate tokenizations, sum MI for each candidate, and rule out all but one candidate. A to-

[4] As an example, for 'cat', the complete list of overlapping n-grams would be 'c', 'a', 't', 'ca', 'at', and 'cat'.

kenization is a candidate if it exhaustively parses the entire string and has no overlapping tokens. Thus, for 'comingle', co+mingle, coming+le, comingle, c+o+m+i+n+g+l+e, etc., are some of the candidates, but comi+ngl and com+mingle are not. To obtain MI, we need to compute the log probability (logp) of every n-gram in the corpus. If S_k (k = 1, …, K) denotes the set of all n-grams of length k, and s_n is a particular n-gram of length n, then we compute logp for s_n as:

$$\text{logp} = \log F(s_n) - \log \Sigma (F(S_n)) \qquad (4)$$

where $F(s_n)$ is the frequency of s_n in the corpus, and $\Sigma (F(S_n))$ is the sum of the frequencies of all S_n in the corpus.[5] In all cases, logp is negative, and MI is maximized when the magnitude of the sum of logp for all elements in the tokenization (also negative) is minimized, i.e. closest to zero. Tokenizations consisting of one, two or more elements (respective examples are comingle, co+mingle, and co+ming+le) will all receive a score, although those with fewer elements will tend to be favored.

We considered some minor variants in the settings for this approach in which word-initial and word-final n-grams were indexed separately from word-medial n-grams. Guided by McNamee and Mayfield's (2004) finding that there is an optimal (language-dependent) value of k for S_k, we also varied the maximum length of n-grams allowed in tokenizations. Under all settings, we followed steps 3-5 from Table 3 (including SVD) from here on.

This approach (which we call latent *morpho*-semantic analysis), then, is like LSA, except that the types and tokens are statistically-derived morphemes rather than terms. Whatever LMSA variant is used, the underlying approach to morphological tokenization is completely language-independent. Example output is shown in Table 5 for wordforms from the Russian lemma пресмыкаться 'to crawl', where the common stem (or at least an approximation thereof) is correctly identified.

Wordform	**Tokenization**	
пресмыкающемуся	пресмыкаю	щемуся
пресмыкающимися	пресмыкаю	щимися
пресмыкающимся	пресмыкаю	щимся
пресмыкающихся	пресмыкаю	щихся

Table 5. Examples of MI-based tokenization

[5] Note that (4) is closely related to the 'weighted mutual information' measure used in Goldsmith (2001: 172).

We do not directly test the accuracy of these tokenizations. Rather, measures of CLIR precision (described in section 4) indirectly validate our morphological tokenizations.

4 Testing framework

To assess our results on a basis comparable with previous work, we used the same training and test data as used in Chew et al. (2007) and Chew and Abdelali (2008). The training data consists of the text of the Bible in 31,226 parallel chunks, corresponding generally to verses, in Arabic, English, French, Russian and Spanish. The test data is the text of the Quran in the same 5 languages, in 114 parallel chunks corresponding to suras (chapters).

Questions are sometimes raised as to how representative the Bible and/or Quran are of modern language. However, there is little question that the number and diversity of parallel languages covered by the Bible[6] is not matched elsewhere (Resnik et al. 1999), even by more mainstream parallel corpora such as Europarl (Koehn 2002)[7]. The diversity of languages covered is a particularly important criterion for our purposes, since we would like to look at methods which enhance retrieval for languages across the analytic-synthetic spectrum. The Bible also has the advantage of being readily available in electronic form: we downloaded all our data in a tab-delimited, verse-indexed format from the 'Unbound Bible' website mentioned above (Biola University, 2005-2006).

In accordance with previous work, we split the test set into each of the 10 possible language-pair combinations: AR-EN, AR-FR, EN-FR, and so on. For each language pair and test, 228 distinct 'queries' were submitted – each query consisting of one of the 228 sura 'documents'. To assess the aggregate performance of the framework, we used average precision at 1 document, hereafter 'P1' (1 if the translation of the document ranked highest, zero otherwise – thus, a fairly strict measure of precision). We also measured precision on a basis not used by Chew et al. (2007) or Chew and Abdelali (2008): multilingual precision at 5 documents (hereafter 'MP5'). For this,

[6] At December 31, 2006, complete translations existed in 429 languages, and partial translations in 2,426 languages (Bible Society 2007).

[7] Since the Europarl text is extracted from the proceedings of the European Parliament, the languages represented are generally closely-related to one another (most being Germanic or Romance).

each of the 570 documents (114 suras, each in 5 languages) is submitted as a query. The results are drawn from the pool of all five languages, so MP5 represents the percentage, on average, of the top 5 documents which are translations of the query. This measure is still stricter than P1 (this is a mathematical necessity) because the retrieval task is harder. Essentially, MP5 measures how well similar documents cluster across languages, while P1 measures how reliably document translations are retrieved when the target language is known.

5 Results and Discussion

The following tables show the results of our tests. First, we present in Table 6 the results using standard LSA, in which terms are sequences of characters delimited by non-word characters. Here, in essence, we reperformed an experiment in Chew and Abdelali (2008).

P1 (overall average: 0.8796)					
	AR	EN	ES	FR	RU
AR	1.0000	0.7544	0.7193	0.7368	0.7544
EN	0.7719	1.0000	0.9123	0.9386	0.9474
ES	0.6316	0.9298	1.0000	0.9298	0.8947
FR	0.7719	0.9035	0.9298	1.0000	0.9386
RU	0.7719	0.9298	0.9035	0.9211	1.0000
MP5: AR 0.4456, EN 0.7211, ES 0.6649, FR 0.7614, RU 0.6947; **overall average: 0.6575**					

Table 6. Results with standard LSA

Our results differ from Chew and Abdelali's (2008) – our precision is higher – because we use a different value of α in equation (2) above (here, 1.8 rather than 1). Generally, we selected α so as to maximize MP5; discussion of this is beyond the scope of this paper, and not strictly relevant in any case, since we present like-for-like comparisons throughout this section. However, Table 6 shows clearly that our results replicate those previously published, in that precision for Arabic (the most 'synthetic' of the five languages) is consistently lower than for the other four.

The next set of results (in Table 7) is for LSA with SVD of an array in which the rows correspond to all overlapping, but not word-spanning, n-grams of fixed length. The best results here, for n=4, are essentially no better on average than those obtained with standard LSA. However, averaging across languages obscures the fact that results for Arabic have significantly improved (for example, where Arabic documents are used as queries, MP5 is now 0.6205 instead of 0.4456). Still, the fact that average MP5 is essentially unchanged means that this is at the expense of results for other languages.

n =	**Average P1**	**Average MP5**
3	0.8340	0.4951
4	0.8779	0.6761
5	0.8232	0.6365
6	0.6957	0.5197
7	0.5321	0.3986

Table 7. Results with LSA / overlapping n-grams of fixed length

Now we present results in Table 8 where SVD is performed on an array in which the rows correspond to all overlapping, but not word-spanning, n-grams of *any* length (varying maximum length).

n ≤	**Average P1**	**Average MP5**
3	0.8235	0.3909
4	0.9039	0.6256
5	0.9095	0.6839
6	0.8863	0.6716
7	0.8635	0.6470

Table 8. Results with LSA / overlapping n-grams of variable length

Here, the best results (with n<=5) more clearly improve upon LSA: the increases in both P1 and MP5, though each only about 0.03 in absolute terms, are highly significant ($p < 0.005$). Very likely this is related to the fact that when n-grams are used in place of words, the out-of-vocabulary problem is alleviated. But there is quite a high computational cost, which will become apparent in Table 10 and the discussion accompanying it.

A practical advantage of the 'morpheme'-by-document array of LMSA, on the other hand, is that this cost is substantially reduced. This is because, as already mentioned, the vast majority of n-grams are eliminated from consideration. However, does taking this step significantly hurt performance? The results for LMSA presented in Table 9 provide an answer to this.

For P1, the results are comparable to standard LSA when we select settings of $n \leq 7$ (maximum permitted morpheme length) or above. But under the stricter MP5 measure, LMSA not only significantly outperforms standard LSA ($p < 0.001$, at $n \leq 9$); the results are also superior to those obtained under any other method we tested. The improvement in MP5 is comparable to that for P1 – 0.677 to 0.707 – when Chew and Abdelali (2008) use the Darwish Arabic light stemmer to provide input to LSA; our approach, however, has the advantage that it is fully unsupervised.

n ≤	Average P1	Average MP5
4	0.6947	0.4411
5	0.8151	0.6102
6	0.8614	0.6793
7	0.8709	0.6912
8	0.8663	0.6856
9	0.8765	0.6909
10	0.8772	0.6740

Table 9. Results with LMSA[8]

As when n-grams are used without MI, fewer types are out-of-vocabulary: for example, with certain settings for LMSA, we found that the percentage of out-of-vocabulary types dropped from 65% under LSA to 29% under LMSA, and the effect was even more marked for Arabic taken individually (78.5% to 34.4%). This is despite the fact mentioned above that LMSA arrays are more economical than LSA arrays: in fact, as Table 10 shows, 22% more economical (the size of the U matrix output by SVD, used to create vectors for new documents, is determined solely by the number of rows, or types). Note also that both LSA and LMSA are significantly more economical than SVD with overlapping n-grams.

Technique	Rows	Nonzeros
LSA	163,745	2,684,938
LSA with overlapping n-grams (where n ≤ 5)	527,506	45,878,062
LMSA	127,722	3,215,078

Table 10. Comparative matrix sizes

Even the results in Table 9 can still be improved upon. Following McNamee and Mayfield's insight that different length n-grams may be optimal for different languages, we attempted to improve precision further by varying *n* independently by language. For all languages but Arabic, n ≤ 9 seems to work well (either increasing or decreasing maximum n resulted in a drop in precision), but by setting n ≤ 6 for Arabic, P1 increased to 0.8874 and MP5 to 0.7368. As comparison of Table 11 with Table 6 shows, some of the most significant individual increases were for Arabic. It should however be noted that the optimal value for n may be dataset-dependent.

Since n is a *maximum* length (unlike in McNamee and Mayfield's experiments), one might expect that increasing n should never result in a drop in precision. We believe the benefit to limiting the size of n is connected to Brown et al.'s (1992: 470) observation that 'as *n* increases, the accuracy of an n-gram model increases, but the reliability of our parameter estimates, drawn as they must be from a limited training text, decreases'. Effectively, the probabilities used in MI are unrepresentatively high for longer n-grams (this becomes clear if one considers the extreme example of an n-gram the same length as the training corpus).

P1 (overall average: 0.8874)					
	AR	EN	ES	FR	RU
AR	1.0000	0.7895	0.7719	0.7281	0.7807
EN	0.8158	1.0000	0.9298	0.9298	0.9123
ES	0.7807	0.9474	1.0000	0.9123	0.8684
FR	0.7632	0.9035	0.9474	1.0000	0.8947
RU	0.7456	0.9298	0.9298	0.9035	1.0000
MP5: AR 0.5140, EN 0.8035, ES 0.8228, FR 0.8035, RU 0.7404; **overall average: 0.7368**					

Table 11. Best results with LMSA

If setting a maximum value for n makes sense in general, the idea of a lower maximum for Arabic in particular also seems reasonable since Arabic words, generally written as they are without vowels, contain on average fewer characters than the other four languages, and contain roots which are usually three or fewer characters long.

6 Conclusion

In this paper, we have demonstrated LMSA, a linguistically (specifically, morphologically) more sophisticated alternative to LSA. By computing mutual information of character n-grams of non-fixed length, we are able to obtain an approximation to a morpheme-by-document matrix which can substitute for the commonly-used term-by-document matrix. At the same time, because mutual information is based entirely on statistics, rather than grammar rules, all the advantages of LSA (language-independence, speed of implementation and fast run-time processing) are retained. In fact, some of these advantages may be increased since the number of index items is often lower.

Although from a linguist's point of view the theoretical advantages of LMSA may be intrinsically satisfying, the benefit is not confined to the theoretical realm. Our empirical results show that LMSA also brings practical benefits, particularly when performing IR with morphologically complex languages like Arabic. Principally, this seems to be due to two factors: alleviation of the

[8] These results are with the stipulation that word-initial and word-final n-grams are distinguished from word-medial n-grams. We also ran experiments in which this distinction was not made. Detailed results are not presented here; suffice it to say that when word-medial and other morphemes were not distinguished, precision was hurt somewhat (lowering it often by several percentage points).

out-of-vocabulary problem and improvement in the associations made by SVD.

We believe that the results we have presented may point the way towards still more sophisticated types of analysis, particularly for multilingual text. We would like to explore, for example, whether it is possible to use tensor decomposition methods like PARAFAC2 to leverage associations between n-grams, words, documents and languages to still better advantage.

Finally, it is worth pointing out that our approach offers an indirect way to test our statistics-based approach to morphological analysis. The better our 'morphemes' correspond to minimal semantic units (as theory dictates they should), the more coherently our system should work overall. In this case, our final arbiter of the system's overall performance is CLIR precision.

In short, our initial attempts appear to show that statistics-based morphological analysis can be integrated into a larger information retrieval architecture with some success.

Acknowledgement

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

S. Abdou, P. Ruck, and J. Savoy. 2005. Evaluation of Stemming, Query Expansion and Manual Indexing Approaches for the Genomic Task. *Proceedings of TREC 2005*.

M. W. Berry, S. T. Dumais., and G. W. O'Brien. 1994. Using Linear Algebra for Intelligent Information Retrieval. *SIAM: Review* 37, 573-595.

Bible Society. 2006. *A Statistical Summary of Languages with the Scriptures*. Accessed Jan 5 2007 at http://www.biblesociety.org/latestnews/latest341-slr2005stats.html.

Biola University. 2005-2006. *The Unbound Bible*. Accessed Jan 29 2008 at http://www.unboundbible.org/.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. 1992. Class-Based *n*-gram Models of Natural Language. *Computational Linguistics* 18(4), 467-479.

P. Chew and A. Abdelali. 2007. Benefits of the 'Massively Parallel Rosetta Stone': Cross-Language Information Retrieval with over 30 Languages. *Proceedings of the Association for Computational Linguistics*, 872-879.

P. Chew and A. Abdelali. 2008. The Effects of Language Relatedness on Multilingual Information Retrieval: A Case Study With Indo-European and Semitic Languages. *Proceedings of the Workshop on Cross-Language Information Access*.

K. Darwish. 2002. Building a shallow Arabic morphological analyzer in one day. *Proceedings of the Association for Computational Linguistics*, 47-54.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41:6, 391-407.

J. Goldsmith. 2001. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics* 27(2), 153-198.

R. A. Harshman. 1972. PARAFAC2: Mathematical and Technical Notes. *UCLA Working Papers in Phonetics* 22, 30-47.

M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, and K. Stanley. 2005. An Overview of the Trilinos Project. *ACM Transactions on Mathematical Software* 31:3, 397-423.

H. Kashioka, Y. Kawata, Y. Kinjo, A. Finch and E. W. Black. 1998. Use of Mutual Information Based Character Clusters in Dictionary-less Morphological Analysis of Japanese. *Proceedings of the 17th International Conference on Computational Linguistics Vol. 1*: 658-662.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. *Proceedings of the Joint Conference on Human Language Technologies and NAACL*, 48-54.

P. Koehn. 2002. Europarl: a Multilingual Corpus for Evaluation of Machine Translation. Unpublished. Accessed Jan 29 2008 at http://www.iccs.inf.ed.ac.uk /~pkoehn/publications/europarl.pdf.

A. Lavie, E. Peterson, K. Probst, S. Wintner, and Y. Eytani. 2004. Rapid Prototyping of a Transfer-Based Hebrew-to-English Machine Translation System. *Proceedings of the TMI-04*.

P. McNamee and J. Mayfield. 2004. Character *N*-Gram Tokenization for European Language Text Retrieval. *Information Retrieval* 7, 73-97.

P. Resnik, M. Broman Olsen, and M. Diab. 1999. The Bible as a Parallel Corpus: Annotating the "Book of 2000 Tongues". *Computers and the Humanities* 33: 129-153.