Towards a “Universal Dictionary” for Multi-Language Information Retrieval Applications

J. Michael Schultz, Mark Y. Liberman

May 22, 2000

1 Introduction

Recent formal evaluations suggest that simple techniques can accomplish high-quality automatic retrieval from multilingual document sets, at least for some tasks. These techniques are based on per-document word counts — what in the monolingual case are familiarly called “bag-of-words” models — generalized across languages using simple word-by-word translation. However, an important practical impediment to coverage of a large number of languages is the need for translation dictionaries. These are time-consuming and expensive to create by hand, and few language pairs have large enough parallel or comparable text corpora for statistical induction methods to be feasible. In this paper, we show that an appropriate metric for term selection in a monolingual English corpus allows us to define a fairly small list, containing about ten thousand inflected forms or about 7500 lemmas, which works essentially as well (for a particular monolingual document classification evaluation) as an unlimited vocabulary of more than 300,000 word forms does. We suggest that such a list can be taken to form the English axis of a sort of “universal dictionary” for document classification tasks, providing a much more efficient path to the addition of new languages.

If proper names can be treated separately, then the “universal dictionary” becomes even smaller — about 5 thousand terms. Given a new language for which no prior resources of any kind exist, an initial mapping for a term list of this size should require only a few person-weeks of work, even if done entirely by hand. Even smaller lists will still be much better than nothing, if terms are added to the translation dictionary in the order specified by a metric of the type we propose. Useful results should be achieved after only a few hours of work, with performance increasing to an asymptote as the translation dictionary reaches its full size.

Of course, term selection may have to be different for different subject areas. Health records can’t be classified on the basis of a term list derived from a corpus of computer repair manuals. However, term selection is done on the English side only, so that a good general list — perhaps almost deserving the hyperbolic name of "universal dictionary" — can be derived from a very large topically balanced corpus, and more specific lists can be derived from easy-to-get corpora in specific domains.

In order to motivate some of its properties, we situate our "universal dictionary" experiment in the context of a description of our entrants in the TDT-2 and TDT-3 tracking evaluations. Because of the extreme simplicity of our system, we feel that the results of the "universal dictionary" experiment ought to generalize to other approaches as well.

1.1 Multilingual Topic Detection and Tracking

The "tracking" task in the Topic Detection and Tracking (TDT) evaluation [2] starts with one to four seed documents describing an event in the news, and asks for all subsequent documents in a stream of news stories to be classified as to whether or not they are about the cited event. As in many other full-text information retrieval tasks, relatively simple techniques based on per-story word counts work quite well at TDT tracking. Such "bag of words" techniques can be set to operate in a way that combines a miss rate of 5% with a false alarm rate of 0.5% on this task, numbers that are nearly at the point where the inherent uncertainty of the task definition begins to make it hard to measure improvement.

When documents in different languages are added to the picture, simple word-by-word translation allows the same bag-of-words techniques to be applied with little or no change. The necessary "translation" can be produced by applying a conventional Machine Translation (MT) system to the document stream, or by simply selecting one or more target-language terms for each source language term, either in document context or in isolation. TDT experiments with mixed Mandarin and English document streams have shown that simple implementations of such approaches work fairly well, coming within about 30% of monolingual performance on the cost metric for the TDT tracking task (a weighted sum of miss and false alarm rates). This difference is substantially smaller than the differences among algorithms for the monolingual task, and will doubtless be narrowed by on-going research into the improvement of translation dictionaries, the selection of translation equivalents, the treatment of proper names, and so forth.

These results from the TDT evaluations are consistent with the results from the rest of the literature, especially the various TREC evaluations. Bag-of-words measures of document similarity have been shown to work well for many tasks that can be built on top of document-to-document comparison, and simple word-by-word translation can in principle generalize these techniques to multilingual document sets with a modest performance cost.

However, the necessary word-by-word translation still requires a translation dictionary. In the simplest case, this is just a partial function from words in Language X and words in Language Y. Somewhat more complex translation dictionaries involve a relation, so that a given word in X may translate to more than one word in Y; in this case, some estimate of the frequency of different translations may be provided, and this estimate may be modified by the word's context in the Language-X document. There are many alternatives in detail: we might be dealing with word forms, stemmed or lemmatized words, or multi-word phrases; we may try to recognize proper names and transliterate them in a special way; we may try to provide special treatment for other categories such as dates, monetary amounts, and so forth. In the particular case of Mandarin and English, we also can take different approaches to the problem of word segmentation on the Mandarin side.

The results of the TDT tracking evaluations show that even very simple and unsophisticated approaches of this type can work surprisingly well, given a large bilingual dictionary to start with. This was true despite the fact that the coverage of the dictionary was not very good, and its quality of the dictionary was suspect in other ways, as it was derived by simple techniques from freely-available sources that were never intended for any such use.

1.2 Motivation for our experiment

The results of the TDT cross-language experiment, and similar results from the TREC cross-language track, are encouraging indications that solutions to some multi-lingual document retrieval or tracking tasks are within our grasp. However, if we face the problem of performing such a task on a document set that includes a new language for which we have no resources at all, several daunting problems face us. If the new language is richly inflected, then some sort of stemmer or lemmatizer must probably be built – we will ignore this problem for our present purposes. But whatever the structure of the language, we need a translation dictionary.

The TDT experiments made use of Mandarin-Engish translation dictionaries involving on the order of 100K words, derived from a combination of "open" sources that permit free re-distribution. As far as we know, there is only one other language pair for which a free resource in this size range is available. For a dozen or so other languages, one might be able to buy large bilingual dictionaries in electronic form, from which such translation dictionaries might be created (semi-)automatically; or one might be able to buy an MT system of adequate quality for use in automated document comparison tasks. Beyond this point, we face keyboarding or scanning a paper dictionary, for the cases in which a suitable one exists; or constructing a translation dictionary from scratch. All of these options will be expensive and time-consuming. For example, creating a translation dictionary of 100K terms, assuming one minute spent on each translation, is roughly a person-year of work – and producing a translation a minute for a year, 9:00 to 5:00 with an hour for lunch, is a hard job at best.

Investment at this scale for tens or hundreds of languages is not unthinkable, given sufficient incentive, but one is certainly motivated to ask if there is an easier way. Do we really need such a large translation dictionary? Many words that occur in the document set are missing from the translation relation anyway – how damaging would even lower coverage be? Or to put it another way, if we only had the time or money to produce translations for N words, how well could we do for a given value of N?

In addressing such questions, we will get the clearest answers by looking first at the monolingual case. There are many detailed choices to be made in setting up a cross-language experiment, and many of these choices are likely to make a bigger difference in small-vocabulary systems than in large ones. A considerable amount of exploration of the algorithmic space would be necessary to find a local optimum for a small-vocabulary cross-language system, and it may well involve quite different choices than those that are optimal for a large-vocabulary system. Also, the translation dictionary we have used in our Mandarin-English experiments is not of especially high quality, with many missing words and many dubious translations. Thus in selecting vocabulary subsets, we typically find that a substantial fraction of the terms in a given selection are missing from the available translation dictionary, and that the fraction is by no means constant across all ways of selecting a subset of a given size. Thus these random flaws will add considerable noise to small-vocabulary tests.

For all these reasons, we think it is most informative to begin with a monolingual experiment: what is the performance of an English-only TDT tracking system, if its vocabulary is artificially limited to a particular N words? Can we find a way of choosing N words so that the penalty for limiting the vocabulary is minimal?

Given a positive answer to this question, we can pose a second question: how should we configure a small-vocabulary cross-language system so that the cross-language penalty is as small as possible? We do not attempt to answer that question in this present paper.

2 Our TDT tracking algorithm

The similarity metric of our tracking system is based on the *idf*-weighted cosine coefficient described in [7] often referred to as $tf{\cdot}idf$. Using this metric, the tracking task becomes two-fold. The first stage is feature selection: we choose a set of features (words or stems) to represent a given topic. These features might be chosen from a single story or from multiple stories. The second stage is threshold determination: choosing a threshold on the $tf{\cdot}idf$ metric that optimizes the miss and false alarm rates for a particular cost function. In effect, the threshold selection normalizes the $tf{\cdot}idf$ similarity metric across topics.

The cosine coefficient as a document similarity metric has been investigated extensively. Here documents (and queries) are represented as vectors in an n-dimensional space, where n is the number of unique terms in the database. The coefficients of the vector for a given document are the term frequencies (tf) for that dimension. The resulting vectors are extremely sparse and typically high frequency words (mostly closed class) are ignored. The cosine of the angle between two vectors is an indication of vector similarity and is equal to the dot-product of the vectors normalized by the product of the vector lengths.

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

$tf \cdot idf$ (term frequency times inverse document frequency) weighting is an ad-hoc modification to the cosine coefficient calculation which weights words according to their *usefulness* in discriminating documents. Words that appear in few documents are more useful than words that appear in many documents. This is captured in the equation for the inverse document frequency of a word:

$$idf(w) = \log_{10}\left(\frac{N}{df(w)}\right)$$

Where $df(w)$ is the number of documents in a collection which contain word w and N is the total number of documents in the collection.

For our implementation we weighted only the topic vector by idf and left the story vector under test unchanged. This allows us to calculate and fix an idf-scaled topic vector immediately after training on the last positive example story for a topic. The resulting calculation for the similarity measure becomes:

$$sim(a,b) = \frac{\sum_{w=1}^{n} tf_a(w) \cdot tf_b(w) \cdot idf(w)}{\sqrt{\sum_{w=1}^{n} tf_a^2(w)} \cdot \sqrt{\sum_{w=1}^{n} tf_b^2(w)}}$$

2.1 UPENN System Attributes

To facilitate testing, the evaluation stories were loaded into a simple document processing system. Once in the system, stories are processed in chronological order testing all topics simultaneously with a single pass over the data[1] at a rate of approximately 6000 stories per minute on a Pentium 266 MHz machine. The system tokenizer delimits on white space and punctuation (and discards it), collapses case, but provides no stemming. A list of 179 stop words consisting almost entirely of closed-class words was also employed. In order to improve word statistics, particularly for the beginning of the test set, we prepended a retrospective corpus (the TDT Pilot Data [4]) of approximately 16 thousand stories.

2.2 Feature Selection

The *choice* as well as *number* of features (here simply words) used to represent a topic has a direct effect on the trade-off between miss and false alarm probabilities. We investigated four methods of producing lists of features each sorted by their effectiveness in discriminating a topic. This then allowed us to easily vary the number of those features for the topic vectors[2].

[1] In accordance with the evaluation specification for this project [2] no information is shared across topics.

[2] We did not employ feature selection on the story under test but used the text in entirety.

1. Keep all features except for words on the stop list.
2. Relative to training stories, sort words by document count, keeping the n most frequent. This approach has the advantage of finding those words which are common across training stories, and therefore are more general to the topic area, but has the disadvantage of extending poorly from the $Nt = 16$ case to the $Nt = 1$ case.
3. For each story, sort by word count (tf), keeping the n most frequent. While this approach tends to ignore low count words which occur in multiple training documents, it generalizes well from the $Nt = 16$ to the $Nt = 1$ case.
4. As a variant on the previous method we tried adding to the initial n features using a simple greedy algorithm. Against a database containing all stories up to and including the Nt-th training story, we queried the database with the n features plus the next most frequent term. If the separation of on-topic and off-topic stories increased, we kept the term, if not we ignored it and tested the next term in the list. We defined separation as the difference between the average on-topic scores and the average of the 20 highest scoring off-topic documents.

Of the feature selection methods we tried the forth one yielded the best results across varying values of Nt, although only slightly better than the much simpler third method. Occam's Razor prompted us to omit this complication from the algorithm. The DET curves[3] in Figure 1 show the effect of varying the number of features (obtained from method 3) on the miss and false alarm probabilities. The upper right most curve results from choosing the single most frequent feature. Downward to the left, in order are the curves for 5, 10, 50, 150 and 300 features. After examining similar plots from the pilot, training and development-test data sets, we set the number of features for our system to 50. It can be seen that there is limited benefit in adding features after this point.

2.3 Normalization / Threshold Selection

With a method of feature selection in place, a threshold for the similarity score must be determined above which stories will be deemed on-topic, and below which they will not. Since each topic is represented by its own unique vector it cannot be expected that the same threshold value will be optimal across all topics unless the scores are normalized. We tried two approaches for normalizing the topic similarity scores.

For the first approach we calculated the similarity of a random sample of several hundred off-topic stories in order to estimate an average off-topic score relative to the topic vector. The normalized score is then a function of the average on-topic scores of the training stories and the average and standard

[3] See [6] for detailed description of DET curves.

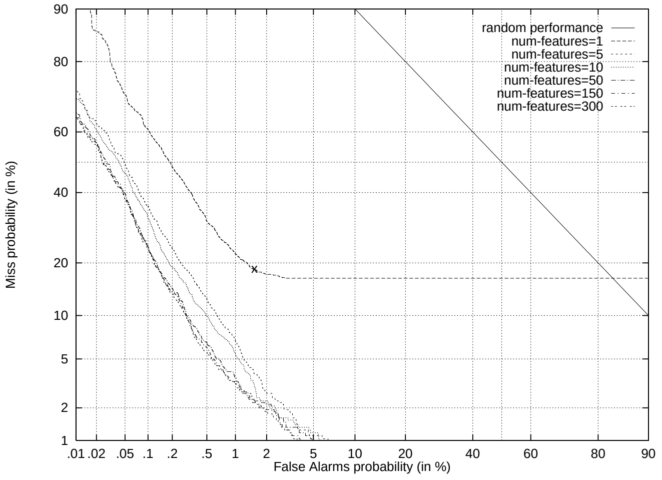

Figure 1: DET curve for varying number of features. (Nt=4, TDT2 evaluation data set, newswire and ASR transcripts)

deviation of the off-topic samples[4]. The second approach looked at only the *highest* scoring *off-topic* stories returned from a query of the topic vector against a retrospective database with the score normalized in a similar fashion to the first approach.

Both attempts reduced the story-weighted miss probability by approximately 10% at low false alarm probability. However, this result was achieved at the expense of higher miss probability at higher false alarm rates, and a higher cost at the operating point defined by the cost function for the task defined in [2].

$$C_{track} = C_{miss} \cdot P(miss) \cdot P_{topic} + C_{fa} \cdot P(fa) \cdot (1 - P_{topic})$$

where

C_{miss} = 1.0 (the cost of a miss)
C_{fa} = 1.0 (the cost of a false alarm, changed to 0.1 in TDT3)
P_{topic} = 0.02 (the *a priori* on-topic probability)

Because of the less optimal trade-off between error probabilities at the point defined by the cost function, we choose to ignore normalization and look directly at cost as a function of a single threshold value across all topics. We plotted $tf{\cdot}idf$ score against story and topic-weighted cost for the training and development-test data sets. As our global threshold we averaged the scores at which story and topic-weighted cost were minimized. This is depicted in figure 2.

[4] σ(on-topic) is unreliable for small Nt but for larger Nt the σ(off-topic) was found to be a good approximation of σ(on-topic).

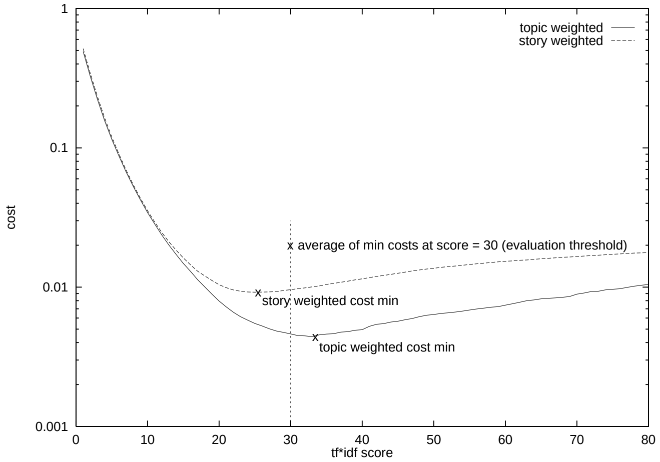

Figure 2: Story and topic-weighted cost as a function of $tf{\cdot}idf$ score. ($Nt = 4$, TDT2 training and development test data sets, newswire and ASR transcripts)

2.4 Tracking Results and Conclusions

We tried a number of approaches to optimize the $tf{\cdot}idf$ weighted cosine coefficient for the tracking task. In the end very simple feature selection with no normalization of topic scores performed as well or better than more sophisticated methods from other sites.

Site	*Story Weighted*		
	$P(miss)$	$P(fa)$	C_{track}
UPenn1	0.0934	0.0040	0.0058
UMass1	0.0855	0.0043	0.0059
BBN1	0.1415	0.0035	0.0063
Dragon1	0.1408	0.0043	0.0070
CMU1	0.2105	0.0035	0.0077
GE1	0.1451	0.0191	0.0216
UMd1	0.8197	0.0062	0.0225
UIowa1	0.0819	0.0492	0.0499

Table 1: Story weighted monolingual tracking results by site. ($Nt = 4$, TDT2 evaluation data set, newswire and ASR transcripts)

Site	*Topic Weighted*		
	$P(miss)$	$P(fa)$	C_{track}
BBN1	0.1185	0.0033	0.0056
UPenn1	0.1092	0.0045	0.0066
Dragon1	0.1054	0.0049	0.0069
UMass1	0.1812	0.0038	0.0074
CMU1	0.2660	0.0023	0.0076
GE1	0.1448	0.0226	0.0251
UMd1	0.6130	0.0156	0.0275
UIowa1	0.1461	0.0425	0.0445

Table 2: Topic weighted monolingual tracking results by site. ($Nt = 4$, TDT2 evaluation data set, newswire and ASR transcripts)

2.5 Generalization to mixed English/Mandarin document sets

For TDT3 we investigated a method of cross-lingual topic tracking built upon our cosine coefficient based monolingual approach. The system relies on a bilingual dictionary for translation as well as for word segmentation in the case of Mandarin. While the system performed above average of those participating in the true bilingual task, in the translated-monolingual[5] task it performed worse than expected. We attribute the poorer than expected results to the difficulty in determining the optimal system threshold but not to the metric's capacity to separate on-topic from off topic stories.

2.6 Topic Tracking in TDT3

In addition to the cross-lingual nature of TDT3, there were a number of other changes in the task definition for tracking[6]. The most substantive change was that no list of off-topic training stories was provided. However, we had already decided for TDT2 to ignore the provided list and rely solely on a independent retrospective corpus for off-topic material. Other changes, which for the most part only affected the relative operating point of the systems, were the decision to the use the topic-weighted score exclusively as the benchmark for system performance (as opposed to story-weighted) and the change to the cost of a false alarm to 0.1 from 1.0 in the cost function. In addition, the cost function was normalized in TDT3 so that a normalized cost of less than one is achieved only when information is extracted from the source data.

$$C_{norm} = C_{track}/min(C_{miss} \cdot P_{target}, C_{fa} \cdot P_{non-target})$$

[5] where the Mandarin text is first translated into English using an MT system.

[6] See [2] for a complete description of TDT3.

The objective of translingual tracking is to identify stories about a particular topic in a target language, given a set of training stories in a source language. As a baseline against which to measure system performance, a corpus was provided with target stories already translated into the source language. This makes it possible to run TDT2 systems over the TDT3 data without any modifications. Although the results from these baseline tests could be submitted as official results, we chose to concentrate on a self-contained system which incorporates the translation aspect of the task using only a bilingual dictionary. The advantage of this approach is that it is more easily applied to other languages than one dependent on a full-blown translation system. However, as the DET curve in Figure 1 shows, it is difficult to approach the performance of the MT based method. The three curves shown all use English as the training language. The worst performing curve is that of the native Mandarin text using our dictionary based approach, next comes the curve representing the stories translated into English using the MT system. Finally, the best results are over the native English text.

2.7 Training Data in the Target Language

Given our decision to incorporate translation into the tracking task, an obvious approach to generating a topic vector in the target language is to simply translate each term in the source vector using a dictionary. We found that a much better approach was to *search* for training stories in the target language during the time period spanned by the training stories of the source language, and then use those stories to generate a topic vector. The advantages of the latter approach are that terms not in the translation dictionary make it into the target vector and also the term counts reflect the native text. The search-based algorithm decreased the system cost by almost half as shown in Table 1.

We searched for training stories among the target language by first translating the source training stories, term for term, into a single large target query. We then $tf{\cdot}idf$ ranked a set of target stories from a time period corresponding to the first and last training story, our logic being that if there are stories to be found in the target language, they should appear during the same time period as those in the source language. From the sorted list, we arbitrarily chose the top ten stories and the query itself to be used as training. At this point we are now in the position to use the monolingual approach of TDT2 over the target language. However, since we track in the target language it is necessary to determine a optimal score threshold for that language. We used the training and development-test portion of the TDT2 corpus to determine a threshold for English and one for Mandarin using the method described in [8] this time optimizing only the topic-weighted cost.

2.8 Word Segmentation in Mandarin

Another aspect of the translingual task, this one particular to Mandarin is word segmentation. Since word boundaries are not explicit in Mandarin text, collecting term statistics based on words is not straight forward. However, on average, word size is approximately 2 characters so collecting overlapping bi-grams is a reasonable approximation to true segmentation. Our segmentation scheme looks for a dictionary entry beginning at the current character of the source text, if an entry is found we segment accordingly. If no entry is found, we create a bi-gram using this and the next character and advance one character in the text. We found using bi-grams where there was no coverage by the dictionary to be more effective than uni-grams in the training data but only slightly more effective in the evaluation data, as is shown in Table 3.

Algorithm	cost
segmentation: dictionary/bi-grams training: translation-based	0.6145
segmentation: dictionary/uni-grams training: search-based	0.3772
segmentation: dictionary/bi-grams training: search-based (used in evaluation system)	0.35305

Table 3: Comparison of algorithms over mandarin only portion of SR=nwt+bnasr TR=eng,nat TE=mul,nat boundary Nt=4

Site	*SR=nwt+bnasr boundary Nt=4*	

	TR=eng,eng
	TE=mul,eng
		TR=eng,nat
		TE=mul,nat

BBN1	0.0922	0.1057
CMU1	0.1376	–
Dragon1	0.1596	–
GE1	0.3778	–
UIowa1	–	0.6051
UMd1	–	0.9662
UPenn1	0.2390 (*Unofficial*)	0.2575

Table 4: Normalized tracking cost by site.

3 The "Universal Dictionary" experiment

In order to select the terms for a "Universal Dictionary", we designed an experiment to investigate the the tradeoff between between tracking cost and vocabulary size for a given metric of term selection. Understanding the relationship between these two parameters will make it possible to build the smallest possible dictionary for a desired level of tracking performance.

Evaluation Condition SR=nwt+bnasr boundary Nt=4	system threshold	optimal threshold
TR=eng,eng TE=mul,eng	0.2390	0.1539
TR=eng,nat TE=mul,nat	0.2575	0.1936
TR=man,nat TE=mul,nat	0.2149	0.1526
TR=mul,nat TE=mul,nat	0.1751	0.1191

Table 5: Cost comparison for system threshold (predicted) vs. optimal threshold (post-hoc)

We began by creating a large dictionary of general English terms using a corpus unrelated to our test corpus[7]. To insure that our approach is not biased toward the time period of the topics, spring of 1998, we chose one half of the 1997 Corpus of North American News [5]. This consists of approximately 250K news stories from two news sources. Using white-space tokenization and without stemming, we collected approximately 300K unique word-forms from these stories.

Next we modified our TDT2 monolingual tracker so that after collecting the terms from the training stories, we remove those not found in the candidate dictionary under test. A topic vector of the 50 most frequent remaining terms was then used for tracking the topic in the same way described in [8]. Our system allowed us to modify the sort criterion and the size of the "universal dictionary" before each run over the test data.

We varied dictionary size from approximately 300 thousand terms down to 100 terms for sorts based on tf, *df*, $tf \cdot idf$, $tf \cdot dpidf$ (to be explained shortly) and, to provide a frame of reference, three random sorts based on different seeds. Figure 3 shows the results of these experiments for the entire range of vocabulary sizes and Figure 4 shows detail for less than 20 thousand terms. In contrast to the actual tracking task where the output is a yes-no decision for each story based on its score relative to a predetermined threshold, here we are interested in only the score itself. The topic-weighted Ctrack cost plotted in these curves represents the theoretical minimum cost of our tracker for a given vocabulary size. Thus Ctrack cost serves our purposes for producing a single value that expresses the performance of vocabulary.

All of the statistics on which the sorts were based come from the North American News Corpus. The first three are well known and obvious first choices for feature selection: term frequency (how many times the word-form occurred), document frequency (how many documents the word-form occurred in), and term frequency weighted by inverse document frequency. The forth and most effective of the sorts is based on term-frequency weighted by the *difference* between a Poisson prediction of *idf* (based on tf) and the actual inverse document frequency. Church and Gale showed in [1] that good keywords for the purposes of IR and categorization tasks often have distributions which differ more from

[7] Our test corpus was the TDT2 evaluation corpus [3] containing 24 test topics

from a Poisson-based expectation than poor ones. Effective keywords tend to "bunch up" in fewer documents than would be expected by a random distribution based on term frequency and the total number of documents. We take Gale and Church's result one step further here by using the difference from Poisson as a weighting for tf for our feature selection.

As figure 4 shows, the advantage of the difference-to-Poisson sort naturally falls off as vocabulary size increases, until, at about 14K terms, it meets with the curves of the other sorts. Table 6 shows the precentual increase in Ctrack cost over an unconstrained vocabulary for the best two sorting metrics. A vocabulary of only 10 thousand terms comes within 8% of the unconstrained vocabulary for the $tf{\cdot}dpidf$ sort. Increasing the vocabulary size to 300K only reduces the increased cost to around 4%.

We examined the 1K vocabularies of the $tf{\cdot}dpidf$ and $tf{\cdot}idf$ based dictionaries and found that of 1000 terms almost 20% (193) differ. In general the quality of the $tf{\cdot}dpidf$ keywords are superior to those of the $tf{\cdot}idf$ sort in the way described by Gale and Church. For example, of the 193 differing terms the $tf{\cdot}dpidf$ dictionary contained about 80 very specific proper nouns whereas the $tf{\cdot}idf$ dictionary contained only about 10 very generic proper nouns (e.g. *ABC, AIDS, Albright, Argentina* vs. *Bob, Calif, February, George*). These proper nouns clearly play an important role in the identification of specific topic areas. Moreover, the more than 100 remaining $tf{\cdot}dpidf$ terms were of much better quality as well (e.g. *accumulate, advertising, aircraft, airline* vs. *able, act, add, ahead*).

Vocabulary size	*% cost increase over unconstrained vocabulary*	
	$tf{\cdot}dpidf$	$tf{\cdot}idf$
1K	254	347
5K	60.4	81.8
7K	17.8	66.9
10K	7.7	15.2
20K	5.4	5.6
300K	4.3	4.3

Table 6: Percentual increase in Ctrack cost over unconstrained vocabulary.

4 Conclusions and Directions for Future Work

Previous work, by ourselves and others, suggests that the penalty for mixed-language document sets in "topic tracking" is no more than about 30% in the TDT cost metric. The new experiment reported here shows that a set of less than 10K words has comparable performance, in the monolingual case, to a full vocabulary of 350K words.

The obvious next step is to combine these two results, and show that a small-vocabulary translation dictionary will allow the mixed-language case to

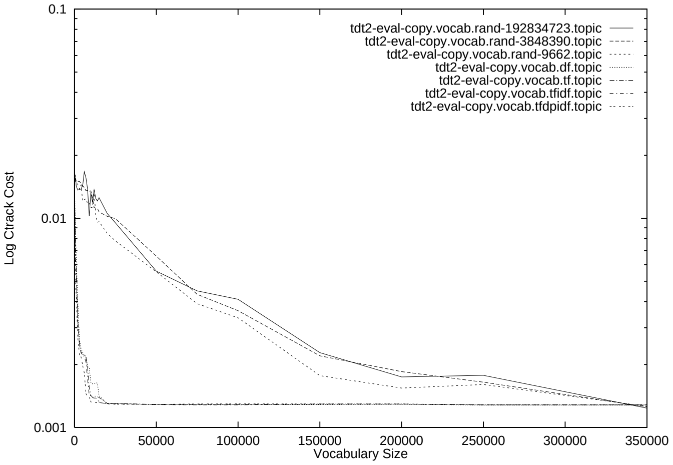

Figure 3: Minumum topic-weighted Ctrack cost vs. vocabulary size for various sorting metrics (overview)

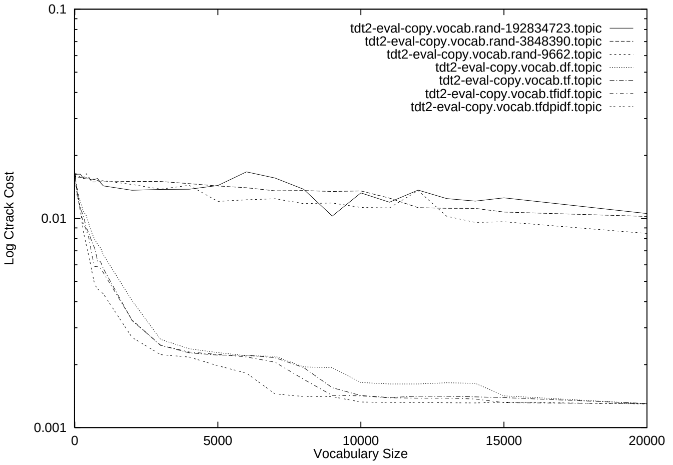

Figure 4: Minimum topic-weighted Ctrack cost vs. vocabulary size for various sorting metrics (detail)

approach monolingual performance. We doubt that simply reducing the vocabulary of our current English/Mandarin system will be a suitable test, because its translation dictionary is of such poor quality. However, the experiment should be tried. A better experiment would be to produce a good-quality translation dictionary for the 10K vocabulary based on the $tf{\cdot}dpidf$ metric, and test it. We plan to do this for a mock-TDT2 experiment in German, a language for which we have a good bilingual dictionary; we may also try to commission a Mandarin translation dictionary of this size.

Other obvious experiments include testing other term-selection metrics, such as mutual information between words and documents; and investigating the effects of treating proper names separately, as names can often be recognized and transliterated dynamically, rather than being stored in a pre-determined list.

References

[1] Kenneth W. Church, William A. Gale, "Inverse Document Frequency (IDF): A Measure of Deviations from Poisson," *Third Workshop on Very Large Corpora*, 1995.

[2] G. Doddington, "The 1999 Topic Detection and Tracking (TDT3) Task Definition and Evaluation Plan" *Available at http://www.nist/gov*, 1999.

[3] G. Doddington, "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan," *Available at http://www.nist.gov/speech/tdt_98.htm*, 1998.

[4] G. Doddington, "The TDT Pilot Study Corpus Documentation," *Available at http://www.ldc.upenn.edu/TDT/Pilot/TDT.Study.Corpus.v1.3.ps*, 1997.

[5] Linguistic Data Consortium, "North American News Text Copus (Suppliment) and AP Worldstream English," *Available from http://www.ldc.upenn.edu*, 1997.

[6] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," *EuroSpeech 1997 Proceedings Volume 4*, 1997.

[7] G. Salton and M.J. McGill, "Introduction to Modern Information Retrieval," *McGraw Hill Book Co.*, New York, 1983.

[8] J. Michael Schultz, Mark Liberman, "Topic Detection and Tracking using idf-weighted Cosine Coefficient," *DARPA Broadcast News Workshop Proceedings*, 1999.