Bilingual Topic Aspect Classification with A Few Training Examples

Yejun Wu and Douglas W. Oard
College of Information Studies and UMIACS Laboratory for Computational Linguistics and Information Processing
University of Maryland, College Park, MD 20742
{wuyj,oard}@umd.edu

ABSTRACT

This paper explores topic aspect (i.e., subtopic or facet) classification for English and Chinese collections. The evaluation model assumes a bilingual user who has found documents on a topic and identified a few passages in each language on aspects of that topic. Additional passages are then automatically labeled using a k-Nearest-Neighbor classifier and local (i.e., result set) Latent Semantic Analysis. Experiments show that when few training examples are available in either language, classification using training examples from both languages can often achieve higher effectiveness than using training examples from just one language. When the total number of training examples is held constant, classification effectiveness correlates positively with the fraction of same-language training examples in the training set. These results suggest that supervised classification can benefit from hand-annotating a few same-language examples, and that when performing classification in bilingual collections it is useful to label some examples in each language.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology— classifier design and evaluation;

General Terms

Performance, Experimentation

Keywords

classification, subtopic, cross-language, test collection

1. INTRODUCTION

We are motivated by the problem of aspectual sentiment characterization: we wish to identify segments of individual documents that address some specified aspect of some specified topic and then characterize the aggregate sentiment expressed about that aspect of that topic in those segments. Because we ultimately wish to do this for multilingual collections, bilingual topic aspect classification is a lesser included problem, and that is the problem on which we focus in this paper. Specifically, we assume that the results of topical search are already available in two languages, and that sentiment analysis will be performed in a subsequent processing stage; our focus is therefore on labeling examples of aspects in the two languages and on training classifiers that can accurately identify additional document segments that address those same aspects in the two languages.

Cross-language text classification problems arise in two settings: (1) training examples have already been labeled in one language, and we wish to build a classifier for another language using that training data, or (2) no training data yet exists, and classifiers must be built that operate well in more than one language. Our focus here is on the second of these settings (although our results offer some insight into the first as well). In particular, we are interested in the specific problem of topic aspect classification, where we mean 'aspect' in the same sense as was used in the TREC Interactive Track (i.e., a facet or specific subtopic of a topic[1]).

It is already known that training examples in one language can be used to build a classifier for another language [2, 5, 16, 17] (indeed, this is the key insight that motivates work on cross-language information retrieval). Our new question in this paper is whether examples from two languages can productively be used together to improve classification effectiveness. Although by no means obvious (since systematic translation errors could equally well have reduced classification accuracy), it turns out that the answer to that question is yes. These results suggest that balancing the investment in annotation of training examples across languages can be helpful when seeking to simultaneously optimize classification effectiveness for more than one language. The paper is organized as follows: Section 2 introduces related work, Section 3 describes our methods for aspect classification, Section 4 addresses the design of the test collection, Section 5 presents our results, and Section 6 concludes the paper.

2. RELATED WORK

The goal of text classification is to classify the topic or theme of a document [10]. Automated text classification is a supervised learning task, defined as automatically assigning pre-defined category labels to documents [23]. It is a well studied task, with many effective techniques. Feature selection is known to be important. The purpose of feature

[1]In linguistics, 'aspect' often denotes 'grammatical aspect;' we consistently mean 'topic aspect.'

SIGIR'08, July 20–24, 2008, Singapore.

selection is to reduce the dimensionality of the term space since high dimensionality may result in the overfitting of a classifier to the training data. Yang and Pedersen studied five feature selection methods for aggressive dimensionality reduction: term selection based on document frequency (DF), information gain (IG), mutual information, a χ^2 test (CIII), and term strength [24]. Using the kNN and Linear Least Squares Fit mapping (LLSF) techniques, they found IG and CIII most effective in aggressive term removal without losing categorization accuracy. They also found that DF thresholding, the simplest method with the lowest cost in computation could reliably replace IG or CIII when the computations of those measure were expensive.

Popular techniques for text classification include probabilistic classifiers (e.g, Naive Bayes classifiers), decision tree classifiers, regression methods (e.g., Linear Least-Square Fit), on-line (filtering) methods (e.g., perceptron), the Rocchio method, neural networks, example-based classifiers (e.g., kNN), Support Vector Machines, Bayesian inference networks, genetic algorithms, and maximum entropy modelling [18]. Yang and Liu [23] conducted a controlled study of 5 well-known text classification methods: support vector machine (SVM), k-Nearest Neighbor (kNN), a neural network (NNet), Linear Least-Square Fit (LLSF) mapping, and Naive Bayes (NB). Their results show that SVM, kNN, and LLSF significantly outperform NNet and NB when the number of positive training examples per category are small (fewer than 10).

In monolingual text classification, both training and test data are in the same language. Cross-language text classification emerges when training data are in some other language. There have been only a few studies on this issue. In 1999, Topic Detection and Tracking (TDT) research was extended from English to Chinese [21]. In *topic tracking*, a system is given several (e.g., 1-4) initial seed documents and asked to monitor the incoming news stream for further documents on the same topic [4], the effectiveness of cross-language classifiers (trained on Chinese data and tested on English) was worse than monolingual classifiers.

Bel et al. [2] studied an English-Spanish bilingual classification task for the International Labor Organization (ILO) corpus, which had 12 categories. They tried two approaches—a *poly-lingual approach* in which both English and Spanish training and test data were available, and *cross-lingual approach* in which training examples were available in one language. Using the *poly-lingual approach*, in which a single classifier was built from a set of training documents in both languages, their Winnow classifier, which, like SVM, computes an optimal linear separator in the term space between positive and negative training examples, achieved F_1 of 0.811, worse than their monolingual English classifier (with F_1=0.865) but better than their monolingual Spanish classifier (with F_1=0.790). For the *cross-lingual approach*, they used two translation methods—*terminology translation* and *profile translation*. When trained on English and tested on Spanish translated into English, their classifier achieved F_1 of 0.792 using *terminology translation* and 0.724 using *profile translation*; when trained on Spanish and tested on pseudo-Spanish, their classifier achieved F_1 of 0.618; all worse than their corresponding monolingual classifiers.

Rigutini et al. [17] studied English and Italian cross-language text classification in which training data were available in English and the documents to be classified were in Italian. They used a Naive Bayes classifier to classify English and Italian newsgroups messages of three categories: *Hardware*, *Auto* and *Sports*. English training data (1,000 messages for each category) were translated into Italian using *Office Translator Idiomax*. Their cross-language classifier was created using Expectation Maximation (EM), with English training data (translated into Italian) used to initialize the EM iteration on the unlabeled Italian documents. Once the Italian documents were labeled, these documents were used to train an Italian classifier. The cross-language classifier performed slightly worse than monolingual classifier, probably due to the quality of their translated Italian data.

Gliozzo and Strapparava [5] investigated English and Italian cross-language text classification by using comparable corpora and bilingual dictionaries (MultiWordNet and the Collins English-Italian bilingual dictionary). The comparable corpus was used for Latent Semantic Analysis which exploits the presence of common words among different languages in the term-by-document matrix to create a space in which documents in both languages were represented. Their cross-language classifier, either trained on English and tested on Italian, or trained on Italian and tested on English, achieved an F_1 of 0.88, worse than their monolingual classifier (with F_1 =0.95 for English and 0.92 for Italian).

Olsson et al. [16] classified Czech documents using English training data. They translated Czech document vectors into English document vectors using a probabilistic dictionary which contained conditional word-translation probabilities for 46,150 word translation pairs. Their "concept label" kNN classifier ($k = 20$) achieved precision of 0.40, which is 73% of the precision of a corresponding monolingual classifier.

The main differences of our approach compared with earlier approaches include: (1) classifying document segments into aspects, rather than documents into topics; (2) using few training examples from both languages; (3) using statistical machine translation results to map segment vectors from one language into the other.

3. METHODS

The goal of bilingual aspect classification is to classify English and Chinese document segments that address the same broad topic based on relevance to specific aspects of that topic. Here we define *monolingual aspect classification* as a task which uses training examples in one language only, and the test examples are in the same languages; we define *bilingual aspect classification* as a task which uses training data in two languages. In this section, we discuss general approaches to monolingual and bilingual aspect classification, classification methods, and evaluation metrics.

3.1 Monolingual Aspect Classification

Our monolingual aspect classification system, which serves as a baseline, was built with 3 steps:

(1) A user who can read and write both English and Chinese retrieves a set of English document segments relevant to a topic from an English collection, and a set of Chinese document segments relevant to the same topic from a Chinese collection. The Indri search engine[2] was used to create the two information retrieval systems. The user then examines the two sets of retrieved document segments and, for each aspect, selects 2–4 document segments from each language.

(2) Local latent semantic analysis (LSA) is performed on

[2]http://www.lemurproject.org/indri/

each set of retrieved document segments to reduce the dimensionality of the term space. In the vector space model, documents (and queries) are represented as term vectors in a t-dimensional space (t is the number of terms) [1], which represents both "signal" (i.e., meaning) and "noise" (from term usage variations). LSA reduces the dimensionality of the vector space with semantic information (hopefully) preserved but conflating similar terms towards a "conceptual" representation. The dimensions that are kept are those that explain the most variance. The mathematical basis for LSA is a Singular Value Decomposition (SVD) of the high-dimensional term-document matrix. The SVD represents both terms and documents as vectors in a space of choosable dimensionality [3]. This yields an optimal approximation to the original term-document matrix in the least squares (or L_2 norm) sense. LSA for a large document collection is both computationally expensive and memory intensive, but local LSA is applied to smaller matrices and thus does not suffer from these computational problems. Local LSA is defined as applying SVD to a term-by-document matrix consisting only of the documents relevant to a topic [7]. The SVD decomposes a rectangular matrix of terms by documents (t $\times$ d) into three matrices. For example, a t $\times$ d matrix of terms and documents X can be decomposed into the product of three other matrices: $X = T_0 S_0 D_0^T$, such that T_0 and D_0 are orthonormal matrices of left and right singular vectors and S_0 is the diagonal matrix of singular values. The diagonal elements of S_0 are constructed to be non-negative and ordered in decreasing magnitude [3]. By choosing the first k largest singular values in S_0 and setting the remaining smaller ones to zero (and deleting the corresponding columns of T_0 and D_0), we get a matrix $\hat{X}$ which is approximately equal to X, but with rank k. The chosen k for our experiments is introduced in Section 5.

(3) A classification algorithm takes the manually selected document segments as training examples and identifies which of the unlabeled document segments on that topic best match that aspect (in reduced term space).

The keys to the classification algorithm are a segment-segment similarity function and a threshold for making the classification decision. In the vector space model, a document segment is represented as a vector of term weights, and the similarity of two document segments can be computed as the cosine of the two vectors. A better index term weighting function can lead to a substantial improvement in information retrieval performance. Okapi BM25 term weighting [14, 19] has been shown to be robust and to achieve retrieval effectiveness that is on a par with any other known technologies, so we compute similarity on Okapi BM25 term weights in local LSA space.

3.2 Bilingual Aspect Classification

Since the user has selected training examples from two languages for a same aspect, the training examples in one language might be used as additional training examples for the other language (if we know how to map them correctly). The process of bilingual aspect classification involves the following steps (introduced using English as the language of the segments to be classified, Chinese as the other language, and translating Chinese into English; but it works for the other direction in a similar way): (1) Once the Chinese aspect training examples are provided by the user, they are translated into English. (2) Fold in (or map) the translated training examples into the original English document segments' LSA space. (3) We suspect that systematic translation errors might put the translated training examples in the wrong place, so optionally, correct the mapping of the translated training examples by moving the centroid of these translated segments toward the centroid of the original English training examples. (4) Classify the unlabeled English segments using the English and the translated Chinese training examples in their English document segments' LSA space.

"Translation" here means mapping term statistics from one language to another, not simply replacing the terms themselves. If a translation probability matrix which estimates the probabilities that Chinese words will be translated into English words is available, a Chinese segment vector can be translated into an English segment vector by multiplying the segment vector by the translation probability matrix, and then folded into an English LSA space by multiplying the resulting English segment vector by the term-by-dimension matrix left singular vector T_0. Translation probabilities can be estimated from parallel corpora, from multilingual dictionaries (when presentation order encodes relative likelihood of general usage), or from the distribution of an attested translation in multiple sources of translation knowledge [20]. In statistical machine translation (MT), translation probabilities are usually learned from parallel corpora. Parallel corpora consist of pairs of documents in two languages that are translations of each other. Sentence-aligned parallel corpora are required for statistical MT. With sentence-aligned parallel corpora, the freely available GIZA++ toolkit [13] can be used to train translation models. GIZA++ produces a representation of a sparse translation matrix. We re-used an existing translated model in our experiment (see Section 5). The document vectors extracted from the English index of a search engine (i.e. Indri in our experiments) are English word stems and their term weights, and a word stem could have resulted from multiple word forms, so we conflated the probabilities of English words into probabilities for their corresponding stems in the probability tables.

3.3 Classification Methods

Previous studies show that k-Nearest-Neighbor (kNN) and Support Vector Machine (SVM) technologies are among the best text classifiers [23]. Since a topic can have multiple aspects, our classification problem is an m-way multiple-class problem. kNN is a natural choice for a multiple-class problem, so we selected kNN for our experiments. Here we introduce the classical kNN approach and two variants. The classical kNN algorithm is very simple: to classify a new object (i.e., a document segment), consult the k training examples that are most similar, where k is an integer, k $\geq$ 1. Each of the k labeled neighbors "votes" for its category. Then count the number of votes each category gets, and assign the category with the maximum number of votes to the new object [10]. In our experiments, the similarity measure was the cosine similarity function with Okapi weights. The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by using empirical techniques (e.g., cross-validation).

In the classical kNN algorithm, the degree of similarity between a test object and training examples is indirectly used

(to pick the k nearest neighbors). To make better use of the similarity scores, we introduce two variants of the classical algorithm that directly use the similarity scores. Franz's algorithm [15] sums up the weighted similarity scores as the contribution of the k nearest neighbors to their categories. That is, each category of the k nearest neighbors accumulates the similarity between the new object and the training examples of this category, the category with the maximum sum of similarity scores is assigned to the new object. A second variant of the classical kNN algorithm was proposed by Yang [22]. Here the conventional M-way classification kNN is adapted to the 2-way classification problem. Since we have multiple aspect classes to work with; when we are working with *Aspect X*, we classify documents either into *Aspect X* or *Non-Aspect X*. Since we have different numbers of positive and negative training examples, we compute a relevance score for each test document as follows [22]:

$$r(x, kp, kn) = \frac{1}{|U_{kp}|} \sum_{y \in U_{kp}} cos(x, y) - \frac{1}{|V_{kn}|} \sum_{z \in V_{kn}} cos(x, z) - T$$

where U_{kp} consists of the kp nearest neighbors of test document segment vector x among the positive training examples, and V_{kn} consists of the kn nearest neighbors of x among the negative training examples; T is a threshold. In our experiments, we set T to 0. In classification, two types of decisions can be made: a *soft* classification decision in which the test object can be assigned to more than one class, and a *hard* classification decision in which the test object has to be assigned to only one class. In our experiments, we elected to make *hard* classification decisions to make the classifiers easier to build—the category with the maximum positive relevance value in Yang's variant or the maximum sum of similarity scores in Franz's variant is assigned. Soft classification, however, can be useful when users define conceptually overlapped aspects or assign the same or overlapping segments to different aspects as training examples.

4. TEST COLLECTION DESIGN

We adopted precision, recall, and the F_1-measure (which we refer to generically as effectiveness). We also reported the standard deviation (stdev) of F_1. We did not use accuracy because accuracy is often dominated by the count of correctly classifying a truly negative test instance into a negative category when the negative category is large. We are interested in how well we do on an aspect, so we used macroaveraged measures [9]. To annotate aspects, we need topics, documents, and a definition of document segments. We have English and Chinese news articles from the Topic Detection and Tracking (TDT) collection, which includes pre-annotated topics: the TDT3 collection with 1999 and 2000 evaluation topics, and the TDT4 collection with 2002 and 2003 evaluation topics. TDT3 topics are described in both English and Chinese languages, whereas TDT4 topics are described only in English.[3] The TDT3 and TDT4 collections include news articles and automatically transcribed broadcast news from 13 news sources. TDT3 and TDT4 also include annotated relevant documents for the topics. Most of the English relevant documents are from NYT, APW, and VOA, whereas most of the Chinese relevant documents are from XIN, ZBN, and VOM (VOA Mandarin); adding other sources does not significantly increase the number of relevant documents. So we selected the news documents from NYT, APW, VOA, XIN, ZBN, and VOM for our experiments. In our test collection, we have 33,388 Chinese documents and 37,083 English documents from the selected sources.

Our first challenge was to estimate the number of aspects needed to preform statistically significant comparisons. Aspects of a topic may be strongly related, so it is more reasonable to assume that it is the topics that are independent. Assuming precision of aspect classification is normally distributed, according to Cohen's general principle, for *power* $= 0.8$, $\alpha = 0.05$, the effect size $d = 0.5$, and a two-tailed paired-sample t-test, total sample size required would be $N = 32$ topics [6]. Since we want at least 2 aspect categories for each topic, this suggests that at least 64 aspects must be annotated. Beyond simply needing enough topics, we need topics with a sufficient number of relevant documents to yield an adequate number of aspect examples. We would have liked to have had at least 8 segments per aspect (so 4 could be used for training, 4 for test). We therefore selected the 50 TDT3 and TDT4 topics for which at least 15 relevant documents were known.

Before annotation started, we had to choose the granularity (i.e., the text unit) for annotation. To make the test collection more broadly useful, we divided each document into sentences and defined a document segment as a group of consecutive sentences. The annotators thereby were free to define the length of a segment in any reasonable manner (e.g., depending on its context). Two graduate students at the University of Maryland were recruited to annotate the aspects. Both were native speakers of Chinese with good mastery of English. A two-step training session was provided before they started the annotation project. In the first step, the first author explained the concept of aspect. An aspect of a topic was defined as a subtopic or a facet of the topic, and an instance of the aspect was defined as a group of consecutive sentences that addressed the aspect. They were provided with an annotated topic as an example, which had been prepared by the first author. In the second step, the annotators were provided with a topic to do a test run to see whether they understood the process. They asked whether "when," "where," and "who" could be defined as aspects of a topic. The first author explained that these were not the aspects we intended because those were too general, and thus could apply to every news topic. They were told that what we intended were more specific and essential subtopics, such as reasons, consequences, people's reactions, and influences on our lives.

The project was split into two phases. In the first phase, each person annotated 25 topics. They were provided with the topics and relevant documents in the two languages, and instructed to identify 2-5 aspects for each topic, and to try to finish annotating each topic within 4 hours. For examining inter-annotator agreement, in the second phase, each person annotated 5 topics that had already been annotated and were recommended by the other person. The two annotators annotated 176 bilingual aspects for the 50 topics, and 50 *All Others* categories which held non-relevant segments from the same documents. The *All Others* categories were incompletely annotated due to time constraints, so they were not used in our experiments. The annotations were automatically examined to remove non-existent sentence numbers, which the annotators occasionally recorded by mistake.

[3]TDT topics: http://projects.ldc.upenn.edu/{TDT3|TDT4}

5. EXPERIMENTS

We designed two experiments. Experiment 1 was designed to test three kNN algorithms and five ways of exploiting foreign-language training examples. Experiment 2 then used the best configuration from Experiment 1 to test the effect of varying the number of same- and foreign-language training examples on classification effectiveness.

TextTiling is a process for automatically subdividing a text document into multi-paragraph "passages" or subtopic segments that are topically coherent [11]. Before we used it to generate document segments, the documents were preprocessed to strip off XML tags. The original paragraph boundaries (marked with <P>) were retained by replacing the <P> tags with two newline characters so that Hearst's TextTiling software recognized them. The TextTiling software has a window size parameter w which defines the length of a text "block." Two text blocks are compared to identify topic shift. Its default value is 20 words. The parameter was optimized to $w = 7$ by running the program on 2,723 relevant documents (for 23 topics) in the NYT/APE/XIE news collection for the TREC 2003 High-Accuracy Retrieval of Documents (HARD) track, with the w parameter ranging from 2 to 28, generating text tiles, then evaluating them with LDC's passage markings. Long tiles were further split if certain conditions were met. If a tile had at least 3 sentences and at least 200 words, or had at least 7 sentences and at least 140 words, it was split into segments with a maximum length of 140 words. These parameters were chosen by manually analyzing some of the long tiles and examining the resulting split tiles when different parameters were tried.

English document segments were indexed using Indri; the Porter stemmer was applied and stopwords removed. Word segmentation for Chinese documents (and topics) was performed with the LDC Segmenter[4] because our translation resources used that segmenter. Since Hearst's TextTiling software does not work with Chinese documents in UTF8 or GB encoding, Chinese documents were converted to hexadecimal codes. Hexadecimal codes for Chinese punctuation were converted into corresponding ASCII. The resulting Chinese document segments were then indexed using Indri; a Chinese stopword list [8] was applied.

TDT topics have 6 components: topic number, topic title, seminal event (What, Who, When, Where), topic explication, rule of interpretation, and examples. We manually prepared queries for each topic using topic titles, *what* and *who* specifications, and topic explications (including "on topic" statements but excluding "off topic" statements). TDT3 topics 30001-30059 already have Chinese versions, other topics were manually translated into Chinese by the first author. We used these queries to retrieve ranked lists of document segments for each language. Next we experimented with how many segments we should use to construct the local LSA space. We had two concerns: (1) we needed enough segments so that most of the segments in the gold standard would contribute to the construction of their local LSA space, and (2) taking too many segments results in slower SVD computation and less potential for dimensionality reduction. We experimented with taking the top 1,000, 1,500, 2,000, 2,500, and 3,000 segments, and ultimately chose the top 1,500 Chinese and 2,500 English segments to construct LSA spaces for those languages (because taking more segments would not have materially increased the number the segments in the gold standard).

Once the number of documents to be retrieved was decided, a term-by-document matrix was constructed (i.e., extracted from the index) for each query. This was the input of the SVD. The outputs we needed were a dimension-by-document matrix D_0, i.e., the document vectors in the LSA space, and a term-by-dimension matrix T_0. Previous study of the relationship between the number of LSA dimensions retained and mean average precision for retrieval from the Cranfield collection of 1,398 aerospace abstracts showed that retaining 100 dimensions yielded good results [12]. Both the number of abstracts and the length of the abstracts in that experiment were close to our case, so we decided to retain 100 dimensions.

We obtained Chinese-English and English-Chinese translation probability tables prepared by Wang [20]. The Chinese-English bidirectional translation probability tables were generated using the Foreign Broadcast Information Service (FBIS) parallel corpus.[5] The word alignment models implemented by GIZA++ are sensitive to translation direction, so GIZA++ was run twice, one with English as the source language and the other with Chinese as the source language [20]. Wang further improved the English to Chinese translation probabilities by combining the translation probabilities in the two directions and applying statistical synonyms derived from the tables [20]. We directly adopted the resulting translation probability matrices.

We could translate a document segment vector in two ways. One was to directly translate Okapi term weights. The other was to extract and translate the TF and DF vectors separately, then to compute the Okapi term weights. We expected that the second way would be better because pre-computed term weights represent the importance of the terms in the source language collection, and the TF*IDF term weight function is not linear—it rewards rare terms. When a rare term was translated from its source language (e.g., Chinese) to the target language (e.g., English), the resulting term weight could be over-estimated. Estimating the importance of the translated terms in the target language (e.g., English) by translating TF and DF vectors separately before computing TF*IDF can avoid this problem.

5.1 Processing the Test Collection

The annotation process resulted in a raw gold standard that could not be directly used for experiments because our systems used machine-generated segments, which were generally different from the hand-annotated segments. We mapped between machine-generated and hand-annotated segments based on sentence overlap. If at least one sentence in a machine-generated segment was marked as relevant to a certain aspect, this segment might be mapped onto that aspect. Therefore, one machine-generated segment could possibly be mapped onto multiple candidate aspects; when this happened, a single mapping decision was made by a majority voting. Ties were resolved arbitrarily by choosing the lowest numbered candidate. We also required that segments be assigned to at most one aspect of a topic. We used greedy selection to ensure that retained aspects were mutually exclusive: if the same sentence appeared in two or more aspects, the first aspect was kept and every other aspect annotated with that sentence was removed. If an aspect in

[4]http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm

[5]LDC catalog: LDC2003E14.

Run	Classical (CL)			Franz			Yang		
	P	R	F_1 (stdev)	P	R	F_1 (stdev)	P	R	F_1 (stdev)
Base	0.506	0.551	0.495 (0.328)	0.536	0.576	0.523 (0.312)	0.553	0.592	0.536 (0.316)
Fold	0.562	0.593	0.536 (0.302)	0.596	0.637	0.582 (0.294)	0.576	0.624	0.563 (0.298)
TrTD	0.572	0.595	**0.539** (0.299)	0.614	0.647	**0.590** (0.272)	0.598	0.638	**0.576** (0.300)
FoldC	0.518	0.517	0.470 (0.296)	0.544	0.542	0.500 (0.293)	0.476	0.498	0.441 (0.294)
TrTDC	0.530	0.539	0.490 (0.318)	0.574	0.565	0.524 (0.293)	0.507	0.530	0.473 (0.292)

Table 1: English Aspect Classification; 4 E and 4 C training examples; mean over 92 aspects from 33 topics.

Runs	Classical (CL)			Franz			Yang		
	P	R	F_1 (stdev)	P	R	F_1 (stdev)	P	R	F_1 (stdev)
Base	0.517	0.535	0.492 (0.305)	0.544	0.544	0.516 (0.314)	0.631	0.621	0.594 (0.288)
Fold	0.544	0.544	0.516 (0.314)	0.618	0.620	0.588 (0.304)	0.627	0.632	0.602 (0.296)
TrTD	0.589	0.576	**0.553** (0.317)	0.637	0.622	**0.599** (0.306)	0.630	0.632	**0.606** (0.305)
FoldC	0.480	0.490	0.446 (0.279)	0.506	0.524	0.482 (0.308)	0.456	0.475	0.432 (0.299)
TrTDC	0.454	0.472	0.436 (0.294)	0.507	0.518	0.484 (0.312)	0.490	0.527	0.473 (0.312)

Table 2: Chinese aspect classification; 4 C and 4 E training examples; mean over 89 aspects from 32 topics.

one language was removed, its corresponding aspect in the other language was also removed. If all aspects of a topic had duplicate sentences, that topic was dropped.

Requiring at least 8 document segments per aspect disqualified too many topics and aspects. We therefore required each aspect to have at least 5 segments, so that 4 segments could be used for training and the others for testing. There were a total of 106 bilingual aspects from 36 topics that met this requirement (excluding the *All Others* categories). To simplify our experiments, we dropped the document segments that were in the gold standard but were not in the ranked list of selected retrieved segments (although we could have kept them by folding them into the LSA spaces). Ultimately we used 92 bilingual aspects from 33 topics, including 3 Chinese aspects that could only be used as training data for English aspect classification because each of them had only 4 segments. This is the first version of our operational test collection.

To examine the effect of varying the number of training examples on classification effectiveness, a second version of the test collection was created in which we required at least 7 segments for each aspect. This version has 40 aspects from 17 topics.

The second phase of the annotation project was a semi-open annotation task because the documents for each aspect were provided, but segments could be freely defined. When validating the annotations for the inter-annotator agreement study, non-existent sentences were removed, but aspects with duplicate sentences were retained. We measured inter-annotator agreement using machine-generated segments that were mapped onto the annotated aspects. In this way we could directly examine the effect of disagreement on our experiment design. The average Cohen's kappa was 0.22 for English and 0.50 for Chinese[6]. The agreement on Chinese aspects was higher than that on English aspects, probably because Chinese was the annotators' native language.

[6]The purpose of annotating recommended topics was to speed up the second phase because annotators often recommended the topics they were most confident in their annotations and they provided clear descriptions for the aspects of the topics. This has a potential bias of reporting a higher inter-annotator agreement.

5.2 Experiment 1

The first experiment was designed to to test whether adding foreign-language training examples would help to classify native-language segments by aspect. We needed as many aspects as possible, so the first version of the test collection was used. The document segments for each aspect were partitioned into training and test sets using cross-validation. We elected to run a maximum of 70 rounds of cross-validation. When foreign-language training segments were added together, the langauge with the greatest number of combinations determined the number of cross-validation rounds. The parameter k of kNN was automatically selected topic by topic depending on the number of aspects for that topic. If a topic had m aspects (excluding the *All Others* category), k was set to $2m + 1$ (an odd integer to minimize ties).

We have 5 ways of using foreign-language training:

Base: baseline, using same-language training only.

Fold: translating foreign-language document segment vectors with pre-computed Okapi term weights, then folding them into the same-language LSA space.

FoldC: translating the foreign-language segment vectors with pre-computed Okapi term weights, folding them into same-language LSA space, then moving them toward the native-language training data so that their centroids meet.

TrTD: translating the foreign-language document segments' TF and DF vectors separately, then computing their Okapi weights, then folding the vectors with Okapi weights into the same-language LSA space.

TrTDC: translating the foreign-language document segments' TF and DF vectors separately, computing their Okapi weights, folding the vectors with Okapi weights into same-language LSA space, then moving them toward the native-language training data so that their centroids meet.

The three classification algorithms (classical kNN, Franz' variant, Yang's variant) were applied to the five ways of using foreign language training data, so there were a total of 15 runs. Experiment 1 tested the effect of adding 4 foreign-

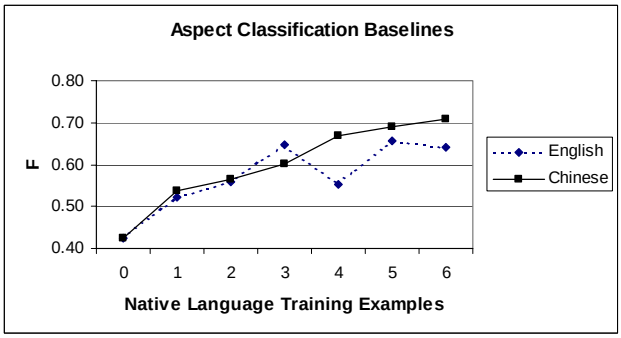

Figure 1: Monolingual case; 40 aspects, 17 topics.

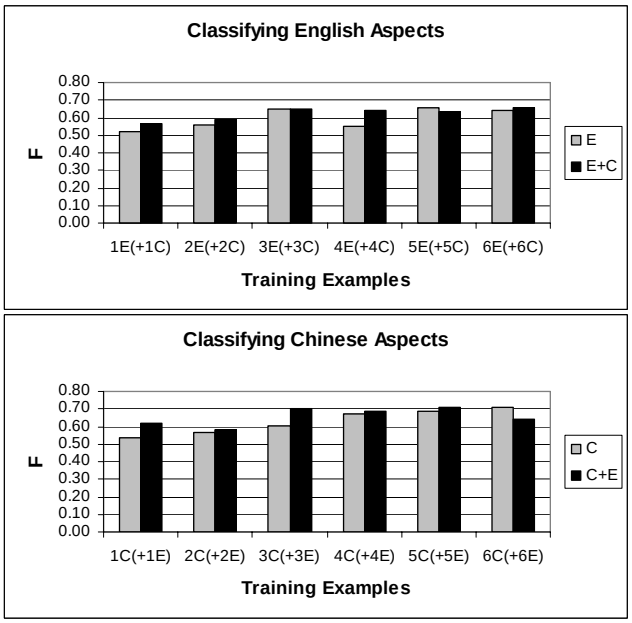

Figure 2: Doubling training with foreign examples.

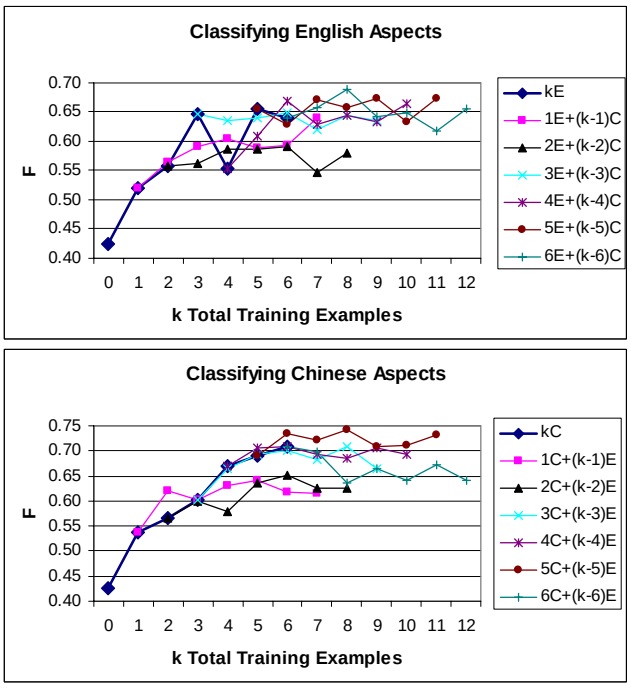

Figure 3: Varying number of foreign examples.

k	Correlation		df
	English	Chinese	
3	0.6391	0.4193	1
4	-0.3832	0.6348	2
5	0.8061*	0.8247*	3
6	0.6347	0.8722*	4
7	0.5829	0.8890*	4
8	0.9253*	0.1729	3
9	0.4374	0.0148	2

Table 3: Effect of same-language fraction on F (k: total examples; *: $p < 0.05$; df: degrees of freedom).

language training examples to 4 same-language training examples on classification performance. Table 1 shows the comparisons of the 15 runs for classifying English aspects. Table 2 shows the similar comparisons for classifying Chinese aspects. Both tables show that both *FoldC* and *TrTDC* consistently yield lower mean precision, recall, and F_1 than *Fold* and *TrTD* respectively, so we focus on comparisons between *Base*, *Fold*, and *TrTD*. *Fold* and *TrTD* consistently improve classification effectiveness over the baseline, indicating that foreign-language training examples are useful. *TrTD* is better than *Fold*, again confirming that translating TF and DF vectors then computing Okapi term weights is better than translating a vector of pre-computed term weights. The two kNN variants are always better than the classical kNN. There is no consistent difference between *Franz TrTD* and *Yang TrTD*. We therefore arbitrarily elected *Franz TrTD* for Experiment 2.

5.3 Experiment 2

The second experiment was designed to examine the effect of varying the number of training examples on classification performance. The second version of the test collection was used, in which each aspect has at least 7 segments in both languages. We used 1-6 document segments for training and the remainder for test (the plotted analytic baseline at zero training examples is based on randomly selecting an aspect for each topic). Again, a maximum of 70 rounds of cross-validations were performed. *Franz TrTD* was used for this experiment. Since fewer than 4 document segments could be used as training examples, the parameter k of kNN was set somewhat differently than in Experiment 1. If an aspect had 1 or 2 training examples, k was the number of training examples; otherwise, k was $2m+1$, where m was the number of aspects for the topic.

Figure 1 shows that when only same-language training examples are involved, more training examples generally yields better classification effectiveness. This meets our expectations. Linear regression models for predicting F from the number of same-language training examples are significant

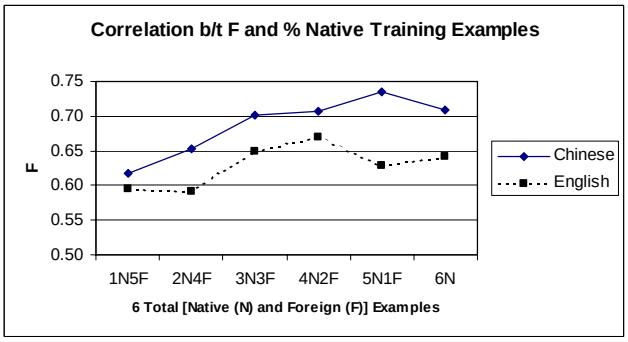

Figure 4: Holding total training examples constant.

at $p < 0.05$ (the correlation coefficient is 0.841 for English, and 0.968 for Chinese). Figure 2 reinforces the conclusion from Experiment 1 that foreign-language training examples are useful. It shows that when equal numbers of foreign language and same-language training examples are used, classification effectiveness usually increases (6C and 5E being the exceptions). Figure 3 shows the effect of supplementing varying numbers of same-language training examples with varying numbers of foreign-language ones. Although a bit busy, it generally shows that a point of diminishing returns is reached beyond which fluctuations appear random. When a fixed number of same-language and foreign-language training examples (k=3-9) are available, classification effectiveness is generally positively correlated with the percentage of same-language training examples. Table 3 shows that Pearson's r correlation scores are positive (except for one outlier), and some are statistically significant at $p < 0.05$. Figure 4 illustrates this correlation graphically for $k = 6$. Due to sparse data, the cases $k < 3$ or $k > 9$ are not included in Table 3.

6. CONCLUSION

Through our experiments, we were able to conclude that when only a few native-language training examples were available, a few additional foreign-language training examples would also be useful. But a point of diminishing returns occurred after a few foreign-langauge training examples were added. when a fixed number of training examples were used, the classification performance was mostly generally correlated with the percentage of native-language training examples. This implies that foreign-language training examples should generally be used as supplements to, rather than substitutes for, native-language training examples.

7. ACKNOWLEDGMENTS

Thanks to Jianqiang Wang for providing us with the bidirectional English-Chinese translation probability tables, and Gina-Anne Levow for proving us with the Chinese stopword list. This work has been supported in part by DARPA contract HR-0011-06-2-0001 (GALE) and NSF award DHB-0729459.

8. REFERENCES

[1] Baeza-Yates, R. and Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. New York: ACM Press.

[2] Bel, N., Koster, C., and Villegas, M., 2003. Cross-lingual text categorization. *ECDL*, 613–620.

[3] Deerwester, S. et al., 1990. Indexing by latent semantic analysis. *JASIS*, 41(6), 391-407.

[4] Franz, M. et al., 2001. Unsupervised and supervised clustering for topic tracking. *SIGIR*.

[5] Gliozzo, A. and Strapparava, C., 2006. Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. *ACL*, 553–560.

[6] Howell, D., 2002. *Statistical Methods for Psychology*, 5th Edition. Pacific Grove, CA: Duxbury.

[7] Hull, D., 1994. *Information Retrieval Using Statistical Classification*. Ph.D. dissertation, Stanford University.

[8] Levow, G., Oard, D, and Resnik, P., 2005. Dictionary-based techniques for cross-language information retrieval. *Information Processing and Management: Special Issue on Cross-language Information Retrieval*, 41(4), 523–547, 2005

[9] Lewis, D., 1991. Evaluating text categorization. *HLT Workshop on Speech and Natural Language*, 312–318.

[10] Manning, C. and Schutze, H., 2000. *Foundations of Statistical Natural Language Processing*, MIT Press.

[11] Hearst, M., 1997. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1), 33–64

[12] Oard, D., 1996. Adaptive vector space text filtering for monolingual and cross-language applications. Ph.D. Dissertation, University of Maryland, College Park

[13] Och, F. and Ney, H., 2000. Improved statistical alighment models. *ACL*, Hong Kong, 440–447.

[14] Olsson, J.S., 2006. An analysis of the coupling between training set and neighborhood sizes for the kNN classifier. *SIGIR'06*, Seattle, Washington. 685-686

[15] Olsson, J.S. and Oard, D., 2007. Improving text classification for oral history archives with temporal domain knowledge. *SIGIR*, Amsterdam.

[16] Olsson, J.S, Oard, D., and Hajic, J., 2005. Cross-language text classification. *SIGIR*, 645–646.

[17] Rigutini, L., Maggini, M., and Liu, B., 2005. An EM based training algorithm for cross-language text classification. *ICWI*, Compiegne, France.

[18] Sebastiani, F., 2002. Machine learning in automated text categorization. *Computing Surveys*, 34(1), 1–47.

[19] Sparck-Jones, K., Walker S. and Robertson, S.E., 2000. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing & Management*, 36(6), 779-840.

[20] Wang, J. and Oard, D., 2006. Combining bidirectional translation and synonym for cross-language information retrieval. SIGIR, Seattle, 202–209.

[21] Wayne, C., 2000. Topic detection and tracking in English and Chinese. *IRAL*, 2000, Hong Kong, 165–172.

[22] Yang, Y., Ault, T., et al. 2000. Improving text categorization methods for event tracking. *SIGIR 2000*.

[23] Yang, Y. and Liu, X., 1999. A re-examination of text categorization methods. *SIGIR 1999*, 42–49.

[24] Yang, Y. and Pedersen, J., 1997. A comparison study on feature selection in text categorization. *ICML*.