Language Identification of Search Engine Queries

Hakan Ceylan
Department of Computer Science
University of North Texas
Denton, TX, 76203
hakan@unt.edu

Yookyung Kim
Yahoo! Inc.
2821 Mission College Blvd.
Santa Clara, CA, 95054
ykim@yahoo-inc.com

Abstract

We consider the language identification problem for search engine queries. First, we propose a method to automatically generate a data set, which uses click-through logs of the Yahoo! Search Engine to derive the language of a query indirectly from the language of the documents clicked by the users. Next, we use this data set to train two decision tree classifiers; one that only uses linguistic features and is aimed for textual language identification, and one that additionally uses a non-linguistic feature, and is geared towards the identification of the language intended by the users of the search engine. Our results show that our method produces a highly reliable data set very efficiently, and our decision tree classifier outperforms some of the best methods that have been proposed for the task of written language identification on the domain of search engine queries.

1 Introduction

The language identification problem refers to the task of deciding in which natural language a given text is written. Although the problem is heavily studied by the Natural Language Processing community, most of the research carried out to date has been concerned with relatively long texts such as articles or web pages which usually contain enough text for the systems built for this task to reach almost perfect accuracy. Figure 1 shows the performance of 6 different language identification methods on written texts of 10 European languages that use the Roman Alphabet. It can be seen that the methods reach a very high accuracy when the text has 100 or more characters. However, search engine queries are very short in length; they have about 2 to 3 words on average, which requires a reconsideration of the existing methods built for this problem.

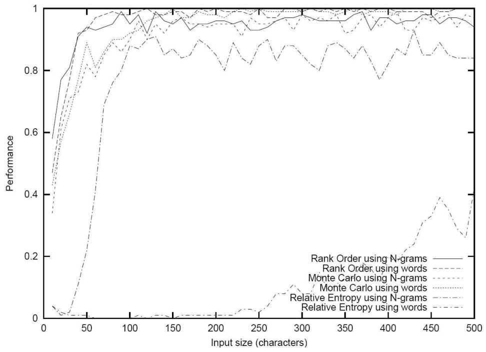

Figure 1: Performance of six Language Identification methods on varying text size. Adapted from (Poutsma, 2001).

Correct identification of the language of the queries is of critical importance to search engines. Major search engines such as Yahoo! Search (www.yahoo.com), or Google (www.google.com) crawl billions of web pages in more than 50 languages, and about a quarter of their queries are in languages other than English. Therefore a correct identification of the language of a query is needed in order to aid the search engine towards more accurate results. Moreover, it also helps further processing of the queries, such as stemming or spell checking of the query terms.

One of the challenges in this problem is the lack of any standard or publicly available data set. Furthermore, creating such a data set is expensive as it requires an extensive amount of work by human annotators. In this paper, we introduce a new method to overcome this bottleneck by automatically generating a data set of queries with language annotations. We show that the data generated this way is highly reliable and can be used to train a machine learning algorithm.

We also distinguish the problem of identifying the textual language vs. the language intended by the users for the search engine queries. For search engines, there are cases where a correct identifi-

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 1066–1074,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP

cation of the language does not necessarily imply that the user wants to see the results in the same language. For example, although the textual identification of the language for the query ”homo sapiens” is Latin, a user entering this query from Spain, would most probably want to see Spanish web pages, rather than web pages in Latin. We address this issue by adding a non-linguistic feature to our system.

We organize the rest of the paper as follows: First, we provide an overview of the previous research in this area. Second, we present our method to automatically generate a data set, and evaluate the effectiveness of this technique. As a result of this evaluation, we obtain a human-annotated data set which we use to evaluate the systems implemented in the following sections. In Section 4, we implement some of the existing models and compare their performance on our test set. We then use the results from these models to build a decision tree system. Next, we consider identifying the language intended by the user for the results of the query, and describe a system geared towards this task. Finally, we conclude our study and discuss the future directions for the problem.

2 Related Work

Most of the work carried out to date on the written language identification problem consists of supervised approaches that are trained on a list of words or n-gram models for each reference language. The word based approaches use a list of short words, common words, or a complete vocabulary which are extracted from a corpus for each language. The short words approach uses a list of words with at most four or five characters; such as determiners, prepositions, and conjunctions, and is used in (Ingle, 1976; Grefenstette, 1995). The common words method is a generalization over the short words one which, in addition, includes other frequently occuring words without limiting them to a specific length, and is used in (Souter et al., 1994; Cowie et al., 1999). For classification, the word-based approaches sort the list of words in descending order of their frequency in the corpus from which they are extracted. Then the likelihood of each word in a given text can be calculated by using rank-order statistics or by transforming the frequencies into probabilities.

The n-gram based approaches are based on the counts of character or byte n-grams, which are sequences of n characters or bytes, extracted from a corpus for each reference language. Different classification models that use the n-gram features have been proposed. (Cavnar and Trenkle, 1994) used an out-of-place rank order statistic to measure the distance of a given text to the n-gram profile of each language. (Dunning, 1994) proposed a system that uses Markov Chains of byte n-grams with Bayesian Decision Rules to minimize the probability error. (Grefenstette, 1995) simply used trigram counts that are transformed into probabilities, and found this superior to the short words technique. (Sibun and Reynar, 1996) used Relative Entropy by first generating n-gram probability distributions for both training and test data, and then measuring the distance between the two probability distributions by using the Kullback-Liebler Distance. (Poutsma, 2001) developed a system based on Monte Carlo Sampling.

Linguini, a system proposed by (Prager, 1999), combines the word-based and n-gram models using a vector-space based model and examines the effectiveness of the combined model and the individual features on varying text size. Similarly, (Lena Grothe and Nrnberger, 2008) combines both models using the ad-hoc method of (Cavnar and Trenkle, 1994), and also presents a comparison study. The work most closely related to ours is presented very recently in (Hammarström, 2007), which proposes a model that uses a frequency dictionary together with affix information in order to identify the language of texts as short as one word.

Other systems that use methods aside from the ones discussed above have also been proposed. (Takci and Sogukpinar, 2004) used letter frequency features in a centroid based classification model. (Kruengkrai et al., 2005) proposed a feature based on alignment of string kernels using suffix trees, and used it in two different classifiers. Finally, (Biemann and Teresniak, 2005) presented an *unsupervised* system that clusters the words based on sentence co-occurence.

Recently, (Hughes et al., 2006) surveyed the previous work in this area and suggested that the problem of language identification for written resources, although well studied, has too many open challenges which requires a more systematic and collaborative study.

3 Data Generation

We start the construction of our data set by retrieving the queries, together with the clicked urls, from the Yahoo! Search Engine for a three months time period. For each language desired in our data set, we retrieve the queries from the corresponding

Yahoo! web site in which the default language is the same as the one sought.[1] Then we preprocess the queries by getting rid of the ones that have any numbers or special characters in them, removing extra spaces between query terms, and lowercasing all the letters of the queries[2]. Next, we aggregate the queries that are exactly the same, by calculating the frequencies of the urls clicked for each query.

As we pointed out in Section 1, and illustrated in Figure 1, the language identification methods give almost perfect accuracy when the text has 100 or more characters. Furthermore, it is suggested in (Levering and Cutler, 2006) that the average textual content in a web page is 474 words. Thus we assume that it is a fairly trivial task to identify the language for an average web page using one of the existing methods.[3] In our case, this task gets already accomplished by the crawler for all the web pages crawled by the search engine.

Thus we can summarize our information in two separate tables; $T1$ and $T2$. For Table $T1$, we have a set of queries Q, and each $q \in Q$ maps to a set of url-frequency pairs. Each mapping is of the form (q, u, f_u), where u is a url clicked for q, and f_u is the frequency of u. Table $T2$, on the other hand, contains the urls of all the web pages known to the search engine and has only two columns; (u, l), where u is a unique url, and l is the language identified for u. Since we do not consider multilingual web pages, every url in $T2$ is unique and has only one language associated with it.

Next, we combine the tables $T1$ and $T2$ using an inner join operation on the url columns. After the join, we group the results by the language and query columns, during which we also count the number of distinct urls per query, and sum their frequencies. We illustrate this operation with a SQL query in Algorithm 1. As a result of these operations, we have, for each query $q \in Q$, a set of triplets $(l, f_l, c_{u,l})$ where l is a language, f_l is the count of clicks for l (which we obtained through the urls in language l), and $c_{u,l}$ is the count of unique urls in language l.

Input: Tables $T1$:[q, u, f_u], $T2$:[u, l]
Output: Table $T3$:[q, l, f_l, $c_{u,l}$]

```
CREATE VIEW T3 AS
SELECT
  T1.q, T2.l, COUNT(T1.u) AS c_{u,l}, SUM(T1.f_u) AS f_l
FROM T1
INNER JOIN T2
ON T1.u = T2.u
GROUP BY q, l;
```

Algorithm 1: Join Tables $T1$ and $T2$, group by query and language, aggregate distinct url and frequency counts.

The resulting table $T3$ associates queries with languages, but also contains a lot of noise. First, we have queries that map to more than one language, which suggests that the users clicked on the urls in different languages for the same query. To quantify the strength of each of these mappings, we calculate a weight $w_{q,l}$ for each mapping of a query q to a language l as:

$$w_{q,l} = f_l / F_q$$

where F_q, the total frequency of a query q, is defined as:

$$F_q = \sum_{l \in L_q} f_l$$

where L_q is the set of languages for which q has a mapping. Having computed a weight $w_{q,l}$ for each mapping, we introduce our first threshold parameter, W. We eliminate all the queries in our data set, which have weights, $w_{q,l}$, below the threshold W.

Second, even though some of the queries map to only one language, this mapping cannot be trusted due to the high frequency of the queries together with too few distinct urls. This case suggests that the query is most likely *navigational*. The intent of navigational queries, such as ”ACL 2009”, is to find a particular web site. Therefore they usually consist of proper names, or acronyms that would not be of much use to our language identification problem. Hence we would like to get rid of the navigational queries in our data set by using some of the features proposed for the task of automatic taxonomy of search engine queries. For a more detailed discussion of this task, we refer the reader to (Broder, 2002; Rose and Levinson, 2004; Lee et al., 2005; Liu et al., 2006; Jansen et al., 2008).

Two of the features used in (Liu et al., 2006) in identification of the navigational queries from click-through data, are the *number of Clicks Satisfied (nCS)* and *number of Results Satisfied (nRS)*. In our problem, we substitute *nCS* with F_q, the total click frequency of the query q, and *nRS* with

[1] We do not make a distinction between the different dialects of the same languge. For English, Spanish and Portuguese we gather queries from the web sites of United States, Mexico, and Brazil respectively.

[2] In this study, we only considered languages that use the Roman alphabet.

[3] Although not done in this study, the urls of web pages that have less than a defined number of words, such as 100, can be discarded to ensure a higher confidence.

U_q, the number of distinct urls clicked for q. Thus we eliminate the queries that have a total click frequency above a given frequency threshold F, and, that have less than a given distinct number of urls, U. Thus, we have three parameters that help us in eliminating the noise from the inital data; W, F, and U. We show the usage of these parameters in SQL queries, in Algorithm 2.

```
Input: Tables $T1$:[$q$, $u$, $f_u$], $T2$:[$u$, $l$], $T3$:[$q$, $l$, $f_l$, $c_{u,l}$]
       Parameters $W$, $F$, and $U$
Output: Table $D$:[$q$, $l$]
---------------------------------------------------------------
CREATE VIEW T4 AS
SELECT T1.$q$, COUNT(T1.$u$) AS $c_u$, SUM(T1.$f_u$) AS $F_q$
FROM T1
INNER JOIN T2 ON T1.$u$ = T2.$u$
GROUP BY q;

CREATE VIEW D AS
SELECT T3.$q$, T3.$l$, T3.$f_l$ / T4.$F_q$ AS $w_{q,l}$
FROM T1
INNER JOIN T4 ON T3.$q$ = T4.$q$
WHERE
T4.$F_q$ < $F$ AND
$w_{q,l}$ >= $W$ AND
T4.$c_{u,l}$ >= $U$;
---------------------------------------------------------------
```

Algorithm 2: Construction of the final data set D, by eliminating queries from $T3$ based on the parameters W, F, and U.

The parameters F, U, and W are actually dependent on the size of the data set under consideration, and the study in (Silverstein et al., 1999) suggests that we can get enough click-through data for our analysis by retrieving a large sample of queries. Since we retrieve the queries that are submitted within a three months period, for each language, we have millions of unique queries in our data set. Investigating a held-out development set of queries retrieved from the United States web site (www.yahoo.com), we empirically decided the following values for the parameters, $W = 1$, $F = 50$, and $U = 5$. In other words, we only accepted the queries for which the contents of the urls agree on the same language, that are submitted less than 50 times, and at least have 5 unique urls clicked.

The filtering process leaves us with 5-10% of the queries due to the conservative choice of the parameters. From the resulting set, we randomly picked 500 queries and asked a native speaker to annotate them. For each query, the annotator was to classify the query into one of three categories:

- **Category-1**: If the query does not contain any foreign terms.

Language	Category-1	Category-1+2	Category-3
English	90.6%	94.2%	5.8%
French	84.6%	93.4%	6.6%
Portuguese	85.2%	93.4%	6.6%
Spanish	86.6%	97.4%	2.6%
Italian	82.4%	96.6%	3.4%
German	76.8%	87.2%	12.8%
Dutch	81.0%	92.0%	8.0%
Danish	82.4%	93.2%	6.8%
Finnish	87.2%	94.0%	6.0%
Swedish	86.6%	95.4%	4.6%
Average	84.3%	93.7%	6.3%

Table 1: Annotation of 500 sample queries drawn from the automatically generated data.

- **Category-2**: If there exists some foreign terms but the query would still be expected to bring web pages in the same language.
- **Category-3**: If the query belongs to other languages, or all the terms are foreign to the annotator.[4]

90.6% of the queries in our data set were annotated as Category-1, and 94.2% as Category-1 and Category-2 combined. Having successful results for the United States data set, we applied the same parameters to the data sets retrieved for other languages as well, and had the native speakers of each language annotate the queries in the same way. We list these results in Table 1.

The results for English have the highest accuracy for Category-1, mostly due to the fact that we tuned our parameters using the United States data. The scores for German on the other hand, are the lowest. We attribute this fact to the highly multilinguality of the Yahoo! Germany website, which receives a high number of non-German queries. In order to see how much of this multi-linguality our parameter selection successfully eliminate, we randomly picked 500 queries from the aggregated but unfiltered queries of the Yahoo! Germany website, and had them annotated as before.

As suspected, the second annotation results showed that, only 47.6% of the queries were annotated as Category-1 and 60.2% are annotated as Category-1 and Category-2 combined. Our method was indeed successful and achieved 29.2% improvement over Category-1, and 27% improvement over Category-1 and Category-2 queries combined.

Another interesting fact to note is the absolute differences between Category-1 and Category-1+2 scores. While this number is very low, 3.8%, for English, it is much higher for the other lan-

[4] We do not expect the annotators to know the etymology of the words or have the knowledge of all the acronyms.

Language	MinC	MaxC	μ_C	MinW	MaxW	μ_W
English	7	46	21.8	1	6	3.35
French	6	74	22.6	1	10	3.38
Portug.	3	87	22.5	1	14	3.55
Spanish	5	57	23.5	1	9	3.51
Italian	4	51	21.9	1	8	3.09
German	3	53	18.1	1	6	2.05
Dutch	5	43	16.3	1	6	2.11
Danish	3	40	14.3	1	6	1.93
Finnish	3	34	13.3	1	5	1.49
Swedish	3	42	13.7	1	8	1.80
Average	4.2	52.7	18.8	1	7.8	2.63

Table 2: Properties of the test set formed by taking 350 Category-1 queries from each language.

guages. Through an investigation of Category-2 non-English queries, we find out that this is mostly due to the usage of some common internet or computer terms such as ”download”, ”software”, ”flash player”, among other native language query terms.

4 Language Identification

We start this section with the implementation of three models each of which use a different existing feature. We categorize these models as statistical, knowledge based, and morphological. We then combine all three models in a machine learning framework using a novel approach. Finally, we extend this framework by adding a non-linguistic feature in order to identify the language intended by the search engine user.

To train each model implemented, we used the EuroParl Corpora, (Koehn, 2005), and the same 10 languages in Section 3. EuroParl Corpora is well balanced, so we would not have any bias towards a particular language resulting from our choice of the corpora.

We tested all the systems in this section on a test set of 3500 human annotated queries, which is formed by taking 350 Category-1 queries from each language. All the queries in the test set are obtained from the evaluation results in Section 3. In Table 2, we give the properties of this test set. We list the minimum, maximum, and average number of characters and words (MinC, MaxC, μ_C, MinW, MaxW, and μ_W respectively).

As can be seen in Table 2, the queries in our test set have 18.8 characters on average, which is much lower than the threshold suggested by the existing systems to achieve a good accuracy. Another interesting fact about the test set is that, languages which are in the bottom half of Table 2 (German, Dutch, Danish, Finnish, and Swedish) have lower number of characters and words on average compared to the languages in the upper half. This is due to the characteristics of those languages, which allow the construction of composite words from multiple words, or have a richer morphology. Thus, the concepts can be expressed in less number of words or characters.

4.1 Models for Language Identification

We implement a statistical model using a character based n-gram feature. For each language, we collect the n-gram counts (for $n = 1$ to $n = 7$ also using the word beginning and ending spaces) from the vocabulary of the training corpus, and then generate a probability distribution from these counts. We implemented this model using the SRILM Toolkit (Stolcke, 2002) with the modified Kneser-Ney Discounting and interpolation options. For comparison purposes, we also implemented the Rank-Order method using the parameters described in (Cavnar and Trenkle, 1994).

For the knowledge based method, we used the vocabulary of each language obtained from the training corpora, together with the word counts. From these counts, we obtained a probability distribution for all the words in our vocabulary. In other words, this time we used a word-based n-gram method, only with $n = 1$. It should be noted that increasing the size of n, which might help in language identification of other types of written texts, will not be helpful in this task due to the unique nature of the search engine queries.

For the morphological feature; we gathered the affix information for each language from the corpora in an unsupervised fashion as described in (Hammarström, 2006). This method basically considers each possible morphological segmentation of the words in the training corpora by assuming a high frequency of occurence of salient affixes, and also assuming that words are made up of random characters. Each possible affix is assigned a score based on its *frequency*, *random adjustment*, and *curve-drop* probabilities, which respectively indicate the probability of the affix being a random sequence, and the probability of being a valid morphological segment based on the information of the preceding or the succeding character. In Table 3, we present the top 10 results of the probability distributions obtained from the vocabulary of English, Finnish, and German corpora.

We give the performance of each model on our test set in Table 4. The character based n-gram model outperforms all the other models with the exception of French, Spanish, and Italian on which the word-based unigram model is better.

English		Finnish		German	
-nts	0.133	erityis-	0.216	-ungen	0.172
-ity	0.119	ihmisoikeus-	0.050	-en	0.066
-ised	0.079	-inen	0.038	gesamt-	0.066
-ated	0.075	-iksi	0.037	gemeinschafts-	0.051
-ing	0.069	-iseksi	0.030	verhandlugs-	0.040
-tions	0.069	-ssaan	0.028	agrar-	0.024
-ted	0.048	maatalous-	0.028	süd-	0.018
-ed	0.047	-aisesta	0.024	menschenrechts-	0.018
-ically	0.041	-iseen	0.023	umwelt-	0.017
-ly	0.040	-amme	0.023	-ches	0.017

Table 3: Top 10 prefixes and suffixes together with their probabilities, obtained for English, Finnish, and German.

The word-based unigram model performs poorly on languages that may have highly inflected or composite words such as Finnish, Swedish, and German. This result is expected as we cannot make sure that the training corpus will include all the possible inflections or compositions of the words in the language. The Rank-Order method performs poorly compared to the character based n-gram model, which suggests that for shorter texts, a well-defined probability distribution with a proper discounting strategy is better than using an ad-hoc ranking method. The success of the morphological feature depends heavily on the probability distribution of affixes in each language, which in turn depends on the corpus due to the unsupervised affix extraction algorithm. As can be seen in Table 3, English affixes have a more uniform distribution than both Finnish and German.

Each model implemented in the previous section has both strengths and weaknesses. The statistical approach is more robust to noise, such as misspellings, than the others, however it may fail to identify short queries or single words because of the lack of enough evidence, and it may confuse two languages that are very similar. In such cases, the knowledge-based model could be more useful, as it can find those query terms in the vocabulary. On the other hand, the knowledge-based model would have a sparse vocabulary for languages that can have heavily inflected words such as Turkish, and Finnish. In such cases, the morphological feature could provide a strong clue for identification from the affix information of the terms.

4.2 Decision Tree Classification

Noting the fact that each model can complement the other(s) in certain cases, we combined them by using a decision tree (DT) classifier. We trained the classifier using the automatically annotated data set, which we created in Section 3. Since this set comes with a certain amount of noise, we pruned the DT during the training phase to avoid overfitting. This way, we built a robust machine learning framework at a very low cost and without any human labour.

Language	Stat.	Knowl.	Morph.	Rank-Order
English	**90.3%**	83.4%	60.6%	78.0%
French	77.4%	**82.0%**	4.86%	56.0%
Portuguese	**79.7%**	75.7%	11.7%	70.3%
Spanish	73.1%	**78.3%**	2.86%	46.3%
Italian	85.4%	**87.1%**	43.4%	77.7%
German	**78.0%**	60.0%	26.6%	58.3%
Dutch	**85.7%**	64.9%	23.1%	65.1%
Danish	**87.7%**	67.4%	46.9%	61.7%
Finnish	**87.4%**	49.4%	38.0%	82.3%
Swedish	**81.7%**	55.1%	2.0%	56.6%
Average	**82.7%**	70.3%	26.0%	65.2%

Table 4: Evaluation of the models built from the individual features, and the Rank-Order method on the test set.

As the features of our DT classifier, we use the results of the models that are implemented in Section 4.1, together with the confidence scores calculated for each instance. To calculate a confidence score for the models, we note that since each model makes its selection based on the language that gives the highest probability, a confidence score should indicate the relative *highness* of that probability compared to the probabilities of other languages. To calculate this relative highness, we use the *Kurtosis* measure, which indicates how peaked or flat the probabilities in a distribution are compared to a normal distribution. To calculate the Kurtosis value, κ, we use the equation below.

$$\kappa = \frac{\sum_{l \in L}(p_l - \mu)^4}{(N-1)\sigma^4}$$

where L is the set of languages, N is the number of languages in the set, p_l is the probability for language $l \in L$, and μ and σ are respectively the mean and the the standard deviation values of $P = \{p_l | l \in L\}$.

We calculate a κ measure for the result of each model, and then discretize it into one of three categories:

- **HIGH**: If $\kappa \geq (\mu' + \sigma')$
- **MEDIUM**: If $[\kappa > (\mu' - \sigma') \wedge \kappa < (\mu' + \sigma')]$
- **LOW**: If $\kappa \leq (\mu' - \sigma')$

where μ' and σ' are the mean and the standard deviation values respectively, for a set of confidence scores calculated for a model on a small development set of 25 annotated queries from each language. For the statistical model, we found $\mu' = 4.47$, and $\sigma' = 1.96$, for the knowledge

Language	500	1,000	5,000	10,000
English	78.6%	81.1%	84.3%	**85.4%**
French	83.4%	85.7%	85.4%	**86.6%**
Portuguese	81.1%	79.1%	**81.7%**	81.1%
Spanish	77.4%	79.4%	81.4%	**82.3%**
Italian	**90.6%**	89.7%	**90.6%**	90.0%
German	81.1%	82.3%	**83.1%**	**83.1%**
Dutch	86.3%	87.1%	**88.3%**	87.4%
Danish	86.3%	87.7%	**88.0%**	**88.0%**
Finnish	88.3%	88.3%	89.4%	**90.3%**
Swedish	81.4%	81.4%	81.1%	**81.7%**
Average	83.5%	84.2%	85.3%	**85.6%**

Table 5: Evaluation of the Decision Tree Classifier with varying sizes of training data.

based $\mu' = 4.69$, and $\sigma' = 3.31$, and finally for the morphological model we found $\mu' = 4.65$, and $\sigma' = 2.25$.

Hence, for a given query, we calculate the identification result of each model together with the model's confidence score, and then discretize the confidence score into one of the three categories described above. Finally, in order to form an association between the output of the model and its confidence, we create a composite attribute by appending the discretized confidence to the identified language. As an example, our statistical model identifies the query *"the sovereign individual"* as English (en), and reports a $\kappa = 7.60$, which is greater than or equal to $\mu' + \sigma' = 4.47 + 1.96 = 6.43$. Therefore the resulting composite attribute assigned to this query by the statistical model is *"en-HIGH"*.

We used the Weka Machine Learning Toolkit (Witten and Frank, 2005) to implement our DT classifier. We trained our system with 500, 1,000, 5,000, and 10,000 instances of the automatically annotated data and evaluate it on the same test set of 3500 human-annotated queries. We show the results in Table 5.

The results in Table 5 show that our DT classifier, on average, outperforms all the models in Table 4 for each size of the training data. Furthermore, the performance of the system increases with the increasing size of training data. In particular, the improvement that we get for Spanish, French, and German queries are strikingly good. This shows that our DT classifier can take advantage of the complementary features to make a better classification. The classifier that uses 10,000 instances gets outperformed by the statistical model (by 4.9%) only in the identification of English queries.

In order to evaluate the significance of our improvement, we performed a paired t-test, with a null hypothesis and $\alpha = 0.01$ on the outputs of the statistical model, and the DT classifier that uses 10,000 training instances. The test resulted in $P = 1.12^{-10} \ll \alpha$, which strongly indicates that the improvement of the DT classifier over the statistical model is statistically significant.

	da	de	en	es	fi	fr	it	nl	sv	pt
da	308	4	9	0	2	3	1	7	14	2
de	7	291	6	2	4	4	5	19	9	3
en	6	8	299	3	3	9	4	5	8	5
es	3	2	4	288	2	2	10	1	1	37
fi	0	5	3	4	316	1	7	4	7	3
fr	2	7	6	3	2	303	10	7	2	8
it	0	1	2	7	4	4	315	2	1	14
nl	5	8	8	4	6	4	4	306	4	1
sv	24	8	6	5	6	2	2	6	286	5
pt	0	1	3	41	1	4	13	2	1	284

Figure 2: Confusion Matrix for the Decision Tree Classifier that uses 10,000 training instances.

In order to illustrate the errors made by our DT classifier, we show the confusion matrix M in Figure 2. The matrix entry M_{l_i,l_j} simply gives the number of test instances that are in language l_i but misclassified by the system as l_j. From the figure, we can infer that, Portuguese and Spanish are the languages that are confused mostly by the system. This is an expected result because of the high similarity between the two languages.

4.3 Towards Identifying the Language Intent

As a final step in our study, we build another DT classifier by introducing a non-linguistic feature to our system, which is the language information of the country from which the user entered the query.[5] Our intuition behind introducing this extra feature is to help the search engine in guessing the language in which the user wants to see the resulting web pages. Since the real purpose of a search engine is to bring the expected results to its users, we believe that a correct identification of the language that the user intended for the results when typing the query is an important first part of this process.

To illustrate this with an example, we consider the query, *"how to tape for plantar fasciitis"*, which we selected among the 500 human-annotated queries retrieved from the United States web site. This query is labelled as Category-2 by the human annotator. Our DT classifier, together with the statistical and knowledge-based models, classifies this query falsely as a Porteguese query, which is most likely caused due to the presence of the Latin phrase *"plantar fasciitis"*.

In order to test the effectiveness of our new feature, we introduce all the Category-2 queries to our

[5] For countries, where the number of official languages is more than one, we simply pick the first one listed in our table.

Language	New Feat.	Classifier-1	Classifier-2
English	74.9%	82.8%	**89.5**%
French	77.0%	85.6%	**93.7**%
Portuguese	79.1%	78.1%	**93.3**%
Spanish	84.1%	80.7%	**94.2**%
Italian	90.6%	86.7%	**96.3**%
German	80.2%	80.7%	**94.2**%
Dutch	91.6%	85.8%	**95.3**%
Danish	88.6%	87.0%	**94.9**%
Finnish	94.0%	87.7%	**97.9**%
Swedish	87.9%	80.9%	**95.3**%
Average	85.0%	83.6%	**94.5**%

Table 6: Evaluation of the new feature and the two decision tree classifiers on the new test set.

test set and increase its size to 430 queries for each language.[6] Then we run both classifiers, with and without the new feature, using a training data size of 10,000 instances, and display the results in Table 6. We also show the contribution of the new feature as a standalone classifier in the first column of Table 6. We labeled the DT classifier that we implemented in Section 4.2 as ”Classifier-1” and the new one as ”Classifier-2”.

Interestingly, the results in Table 6 tell us that a search engine can achieve a better accuracy than Classifier-1 on average, should it decide to bring the results based only on the geographical information of its users. However one can argue that this would be a bad idea for the web sites that receive a lot of visitors from all over the world, and also are visited very often. For example, if the search engine's United States web site, which is considered as one of the most important markets in the world, was to employ such an approach, it'd only receive 74.9% accuracy by misclassifying the English queries entered from countries for which the default language is not English. On the other hand, when this geographical information is used as a feature in our decision tree framework, we get a very high boost on the accuracy of the results for all the languages. As can be seen in Table 6, Classifier-2 gives the best results.

5 Conclusions and Future Work

In this paper, we considered the language identification problem for search engine queries. First, we presented a completely automated method to generate a reliable data set with language annotations that can be used to train a decision tree classifier. Second, we implemented three features used in the existing language identification methods, and compared their performance. Next, we built a decision tree classifier that improves the results on average by combining the outputs of the three models together with their confidence scores. Finally, we considered the practical application of this problem for search engines, and built a second classifier that takes into account the geographical information of the users.

Human annotations on 5000 automatically annotated queries showed that our data generation method is highly accurate, achieving 84.3% accuracy on average for Category-1 queries, and 93.7% accuracy for Category-1 and Category-2 queries combined. Furthermore, the process is fast as we can get a data set of size approximately 50,000 queries in a few hours by using only 15 computers in a cluster.

The decision tree classifier that we built for the textual language identification in Section 4.2 outperforms all three models that we implemented in Section 4.1, for all the languages except English, for which the statistical model is better by 4.9%, and Swedish, for which we get a tie. Introducing the geographical information feature to our decision tree framework boosts the accuracy greatly even in the case of a noisier test set. This suggests that the search engines can do a better job in presenting the results to their users by taking the non-linguistic features into account in identifying the *intended language* of the queries.

In future, we would like to improve the accuracy of our data generation system by considering additional features proposed in the studies of automated query taxonomy, and doing a more careful examination in the assignment of the parameter values. We are also planning to extend the number of languages in our data set. Furthermore, we would like to improve the accuracy of Classifier-2 with additional non-linguistic features. Finally, we will consider other alternatives to the decision tree framework when combining the results of the models with their confidence scores.

6 Acknowledgments

We are grateful to Romain Vinot, and Rada Mihalcea, for their comments on an earlier draft of this paper. We also would like to thank Sriram Cherukiri for his contributions during the course of this project. Finally, many thanks to Murat Birinci, and Seçkin Kara, for their help on the data annotation process, and Cem Sözgen for his remarks on the SQL formulations.

[6] We don't have equal number of Category-2 queries in each language. For example, English has only 18 of them whereas Italian has 71. Hence the resulting data set won't be balanced in terms of this category.

References

C. Biemann and S. Teresniak. 2005. Disentangling from babylonian confusion - unsupervised language identification. In *Proceedings of CICLing-2005, Computational Linguistics and Intelligent Text Processing*, pages 762–773. Springer.

Andrei Broder. 2002. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10.

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

J. Cowie, Y. Ludovic, and R. Zacharski. 1999. Language recognition for mono- and multi-lingual documents. In *Proceedings of Vextal Conference*, Venice, Italy.

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

Gregory Grefenstette. 1995. Comparing two language identification schemes. In *Proceedings of JADT-95, 3rd International Conference on the Statistical Analysis of Textual Data*, Rome, Italy.

Harald Hammarström. 2006. A naive theory of affixation and an algorithm for extraction. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology at HLT-NAACL 2006*, pages 79–88, New York City, USA, June. Association for Computational Linguistics.

Harald Hammarström. 2007. A fine-grained model for language identification. In *F. Lazarinis, J. Vilares, J. Tait (eds) Improving Non-English Web Searching (iNEWS07) SIGIR07 Workshop*, pages 14–20.

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

Norman C Ingle. 1976. A language identification table. *The Incorporated Linguist*, 15(4):98–101.

Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. 2008. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, 44(3):1251–1266.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit, Phuket, Thailand*, pages 79–86.

Canasai Kruengkrai, Prapass Srichaivattana, Virach Sornlertlamvanich, and Hitoshi Isahara. 2005. Language identification based on string kernels. In *In Proceedings of the 5th International Symposium on Communications and Information Technologies (ISCIT-2005*, pages 896–899.

Uichin Lee, Zhenyu Liu, and Junghoo Cho. 2005. Automatic identification of user goals in web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 391–400, New York, NY, USA. ACM.

Ernesto William De Luca Lena Grothe and Andreas Nrnberger. 2008. A comparative study on language identification methods. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Ryan Levering and Michal Cutler. 2006. The portrait of a common html web page. In *DocEng '06: Proceedings of the 2006 ACM symposium on Document engineering*, pages 198–204, New York, NY, USA. ACM Press.

Yiqun Liu, Min Zhang, Liyun Ru, and Shaoping Ma. 2006. Automatic query type identification based on click through information. In *AIRS*, pages 593–600.

Arjen Poutsma. 2001. Applying monte carlo techniques to language identification. In *In Proceedings of Computational Linguistics in the Netherlands (CLIN)*.

John M. Prager. 1999. Linguini: Language identification for multilingual documents. In *HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 2*, page 2035, Washington, DC, USA. IEEE Computer Society.

Daniel E. Rose and Danny Levinson. 2004. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA. ACM.

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

Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. 1999. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12.

C. Souter, G. Churcher, J. Hayes, and J. Hughes. 1994. Natural language identification using corpus-based models. *Hermes Journal of Linguistics*, 13:183–203.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904, Denver, CO.

Hidayet Takci and Ibrahim Sogukpinar. 2004. Centroid-based language identification using letter feature set. In *CICLing*, pages 640–648.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition.