A Language Identification Method Applied to Twitter Data

Anil Kumar Singh
IIT (BHU), Varanasi, India
nlprnd@gmail.com

Pratya Goyal
NIT, Surat, India
goyalpratya@gmail.com

Resumen: Este paper presenta los resultados de varios experimentos que hacen uso de un algoritmo sencillo, guiado por heurísticas, para la finalidad de identificar el idioma en datos de Twitter. Estos experimentos son parte de la tarea compartida que se centra en este problema. El algoritmo se basa en una métrica de distancia calculada a partir de n-gramas. Este algoritmo había sido evaluado satisfactoriamente en textos normales previamente. La métrica de distancia utilizada en este caso es una entropía cruzada simétrica.
Palabras clave: identificación de idioma, entropía cruzada simétrica, microblogging

Abstract: This paper presents the results of some experiments on using a simple algorithm, aided by a few heuristics, for the purposes of language identification on Twitter data. These experiments were a part of a shared task focused on this problem. The core algorithm is an n-gram based distance metric algorithm. This algorithm has previously been shown to work very well on normal text. The distance metric used is symmetric cross entropy.
Keywords: Language identification, symmetric cross entropy, microblogging

1 Introduction and Objectives

Language identification was perhaps the first natural language processing task for which a statistical method was used successfully (Beesley, 1988). Over the years, many algorithms have become available that work very well with normal text (Dunning, 1994; Combrinck and Botha, 1994; Jiang and Conrath, 1997; Teahan and Harper, 2001; Martins and Silva, 2005). However, with the recent spread of social media globally, the need for language identification algorithms that work well with the data available on such media has been felt increasingly. There has been a special focus on microblogging data, because of at least two main reasons. The first is that microblogs have too little data for traditional algorithms to work well directly and the second is that microblogs use a kind of abbreviated language where, for example, many words are not fully spelled out. Some other facts about such data, like multilinguality of many microbloggers only make the problem harder.

Our goal was to take one of the algorithms that has been shown to work very well for normal text, add some heuristics to it, and see how far it goes in performing language identification for microblog data.

2 Architecture and Components of the System

The system we have used is quite simple. There are only two components in the system. At its core there is a language identifier for normal text. The only other module is a preprocessing module. This preprocessing module implements some heuristics. There are two main heuristics implemented. The first one is based on the knowledge that word boundaries are an important source of linguistic information that can help a language processing system perform better. We just wrap every word (more accurately, a token) inside two special symbols, one for word beginning and the other for word ending. The effect of this heuristic is that it not only provides additional information, it also ‘expands’ the short microblogging text a little bit, which is statistically important.

The other heuristic relates to cleaning up the data. Microblogging text, particularly Twitter text, contains extra-textual tokens such as hashtags, mentions, retweet symbols, URLs etc. This heuristic removes such extra-textual tokens from the data before training as well as before language identification.

The intuitive basis of our algorithm is similar to the unique n-gram based approach, which was first used for human identification (Ingle, 1976) and later for automatic identification (Newman, 1987). The insight behind these methods is as old as the time of Ibn ad-Duraihim who lived in the 14th century.

It is worth noting that when n-grams are used for language identification, normally no distinction is made between orders of n-grams, that is, unigrams, bigrams and trigrams etc. are all given the same status. Further, when using vector space based distance measures, n-grams of all orders are merged together and a single vector is formed. It is this vector over which the distance measures are applied.

3 The Core Algorithm

The core algorithm that we have used (Singh, 2006) is an adaptation of the one used by Cavnar and Trenkle (Cavnar and Trenkle, 1994). The main difference is that instead of using the sum of the differences of ranks, we use symmetric cross entropy as the similarity or distance measure.

The algorithm can be described as follows:

1. Train the system by preparing character based and word based (optional) *n*-grams from the training data.
2. Combine *n*-grams of all orders (O_c for characters and O_w for words).
3. Sort them by rank.
4. Prune by selecting only the top N_c character *n*-grams and N_w word *n*-grams for each language-encoding.
5. For the given test data or string, calculate the character *n*-gram based score sim_c with every model for which the system has been trained.
6. Select the t most likely language-encoding pairs (training models) based on this score.
7. For each of the t best training models, calculate the score with the test model. The score is calculated as:

   $$score = sim_c + a * sim_w \qquad (1)$$

   where c and w represent character based and word based *n*-grams, respectively. And a is the weight given to the word based *n*-grams. In our experiment, this weight was 1 for the case when word *n*-grams were considered and 0 when they were not.
8. Select the most likely language-encoding pair out of the t ambiguous pairs, based on the combined score obtained from word and character based models.

The parameters in the above algorithm are:

1. Character based *n*-gram models P_c and Q_c
2. Word based *n*-gram models P_w and Q_w
3. Orders O_c and O_w of *n*-grams models
4. Number of retained top *n*-grams N_c and N_w (pruning ranks for character based and word based *n*-grams, respectively)
5. Number t of character based models to be disambiguated by word based models
6. Weight a of word based models

In our case, for the twitter data, we have not used word based n-grams as they do not seem to help. Adding them does not improve the results. Perhaps the reason is that there is too little data in terms of word n-grams. So the parameters for our case are:

$$O_c = 7,\ O_w = 0,\ N_c = 1000,\ N_w = 0,\ a = 0$$

We used an existing implementation of this algorithm which is available as part of a library called Sanchay[1] (version 0.3.0).

The parameters were selected based on repeated experiments. The ones selected are those which gave the best results. The length of n-grams was selected as 7-grams and we did find that increasing n-gram length improves the results.

In this paper we have used this technique for monolingual identification in accordance with the task definition, but it can be used for multilingual identification (Singh and Gorla, 2007), although the accuracies are not likely to be high when used directly.

4 Resources Employed

For our experiments reported here we have only used the training data provided. We have not used any other resources. We have also, so far, not used any additional tools such as a name entity recognizer. We have implemented some heuristics as described in the previous section.

5 Setup and Evaluation

We evaluated with two different setups. Before the test data for the shared task was released, we had randomly divided the training data into two sets by the usual 80-20 split: one for training and one for evaluation. We also used two evaluation methods.

[1] http://sanchay.co.in

Table 1: Language-wise Results in Percentages (Macroaverages)

	Training 80-20 Split			Test Set		
Language	Precision	Recall	F-measure	Precision	Recall	F-measure
Spanish	91.62	82.05	86.57	93.12	85.93	89.38
Catalan	74.84	84.27	79.28	63.43	81.99	71.52
Portuguese	86.79	73.95	79.86	65.03	88.53	74.98
Galician	34.97	55.34	42.86	25.71	50.12	33.99
Basque	66.67	71.15	68.83	49.30	76.74	60.03
English	80.53	80.53	80.53	71.44	76.53	73.90
Undefined	42.11	16.67	23.88	42.53	7.84	13.24
Ambiguous	1.00	69.62	82.09	1.00	78.08	87.69
Global	72.19	67.20	68.26	63.82	68.25	63.10

One was simple precision based on microaverages, while the other was using the evaluation script provided by the organizers, which was based on macroaverages. Under this setup, on repeated runs, the algorithm described earlier, out of the box, gave a (microaverages based) precision of little more than 70%. On adding the word boundary heuristic to the data, the precision increased to around 78%. On further adding the cleaning heuristic, the precision reached 80.80%. The corresponding macroaverge based F-score was 68.26%.

However, once the test data for the shared task was released and we used it with our algorithm, along with the heuristics, the (macroaverage based) F-score was 61.5%. This increased a little after we slightly improved the implementation of the preprocessing module. The corresponding microaverage based precision was 77.47%. On looking at the results for each language, we find that the performance was best for Spanish (89.38% F-measure) and worst for Galician (33.99% F-measure). These results are presented in table-1.

Tables 2 and 3 list the most frequent single label errors for the two cases (80-20 split of the training data and the test set). While some of the results are as expected, others are surprising. For example, Galician and Portuguese are very similar and they are confused for one another. Similarly for Spanish and Catalan. But it is surprising that Catalan is identified as English and Basque as Spanish. Also, Galician and Portuguese are similar, but the results for them are different. These discrepancies become a little clearer if we notice the fact that the results are quite different in many ways for the two cases: the 80-20 split and the test set. The most probable reason for these discrepancies is that since this method is based purely on distributional similarity, differences in training or testing distributions cause unexpected errors. The fact that there is more data available for some languages (Spanish and Portuguese) and less for others (Galician, Catalan and Basque), the difference being very large, contributes to these discrepancies. It may also be noted that the results were much better in terms of microaverage based precision because in that case our evaluation method took into account multi-label classification such as ‘en+pt’. In fact, each multi-label combination was treated as a single class, both in the case of code switching and ambiguity. As a result, many (around half) of the errors were of such as ‘en’ being identified as ‘en+pt’. This also contributed to making our results lower as evaluated by the script provided by the organizers.

Table 2: Top single label errors on the training 80-20 split

Language	Identified As	No. of Times
Spanish	Catalan	212
Portuguese	Spanish	72
Galician	Portuguese	37
Undef	Basque	31
Catalan	Spanish	29
Basque	Spanish	20
English	Spanish	13
Other	Spanish	6

Table 3: Top single label errors on the test set

Language	Identified As	No. of Times
Spanish	Catalan	1879
Undef	Galician	494
Other	Portuguese	382
Catalan	English	214
Portuguese	Galician	212
Galician	Portuguese	209
Basque	Spanish	59

6 Conclusions and Future Work

We presented the results of our experiments on using an existing algorithm for language identification on the Twitter data provided for the shared task. We tried the algorithm as it is and also with some heuristics. The two main heuristics were: adding the word boundaries to the data in the form of special symbols and cleaning up hashtags, mentions etc. The results were not state of the art for Twitter data (Zubiaga et al., 2014), but they might show how far an out of the box well-performing algorithm can go for this purpose. Also, the results were significantly worse for the test data than they were for the 80-20 split on the provided training data. This means either the algorithm lacks robustness when it comes to microblogging data, or there is a data shift between the training and test data. Perhaps one important conclusion from the experiments is that adding word boundary markers to the data can significantly improve the performance.

For future work, we plan to experiment with techniques along the lines suggested in recent work (Kiciman, 2010; Carter, Weerkamp, and Tsagkias, 2013; Lui and Baldwin, 2014) on language identification for Twitter data.

References

Adams, Gary and Philip Resnik. 1997. A language identification application built on the Java client-server platform. In Jill Burstein and Claudia Leacock, editors, *From Research to Commercial Applications: Making NLP Work in Practice*. Association for Computational Linguistics, pages 43–47.

Beesley, K. 1988. Language identifier: A computer program for automatic natural-language identification on on-line text.

Carter, Simon, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog language identification: overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215.

Cavnar, William B. 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.

Combrinck, H. and E. Botha. 1994. Automatic language identification: Performance vs. complexity. In *Proceedings of the Sixth Annual South Africa Workshop on Pattern Recognition.*

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

Ingle, Norman C. 1976. A language identification table. In *The Incorporated Linguist, 15(4).*

Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy.

Kiciman, Emre. 2010. Language differences and metadata features on twitter. In *Web N-gram Workshop at SIGIR 2010.* ACM, July.

Lui, Marco and Timothy Baldwin. 2014. Accurate language identification of twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 17–25, Gothenburg, Sweden, April. Association for Computational Linguistics.

Martins, Bruno and Mario J. Silva. 2005. Language identification in web pages. In *Proceedings of ACM-SAC-DE, the Document Engeneering Track of the 20th ACM Symposium on Applied Computing.*

Newman, Patricia. 1987. Foreign language identification - first step in the translation process. In *Proceedings of the 28th Annual Conference of the American Translators Association.* pages 509–516.

Simon, Kranig. 2005. Evaluation of language identification methods. In *BA Thesis.* Universitt Tbingens.

Singh, Anil Kumar. 2006. Study of some distance measures for language and encoding identification. In *Proceeding of ACL 2006 Workshop on Linguistic Distances. Sydney, Australia*, Sydney, Australia. Association for Computational Linguistics.

Singh, Anil Kumar and Jagadeesh Gorla. 2007. Identification of languages and encodings in a multilingual document. In *Proceedings of the 3rd ACL SIGWAC Workshop on Web As Corpus*, Louvain-la-Neuve, Belgium.

Teahan, W. J. and D. J. Harper. 2001. Using compression based language models for text categorization. In *J. Callan, B. Croft and J. Lafferty (eds.), Workshop on Language Modeling and Information Retrieval.* ARDA, Carnegie Mellon University, pages 83–88.

Zubiaga, Arkaitz, Iaki San Vicente, Pablo Gamallo, Jos Ramom Pichel, Iaki Alegria, Nora Aranberri, Aitzol Ezeiza, and Vctor Fresno. 2014. Overview of TweetLID: Tweet Language Identification at SEPLN 2014. In *Proceedings of TweetLID @ SEPLN 2014*, Girona, Spain.