LANGUAGE IDENTIFICATION OF TWEETS USING LZW COMPRESSION

Brian O. Bush

Center for Spoken Language Understanding, Oregon Health & Science University, Portland, Oregon

INTRODUCTION

- Goal: Accurate language identification (LID) in several languages for short texts using constrained training resources.
- Language identification is deciding a language among k candidate natural languages a given text is written.
- Our work focuses on short texts, namely, Twitter tweets.
- Benedetto et al. (2002) presented the problem of language identification as simply an extension of Shanons idea of entropy in that the *compressibility of any given document depends on the source language*.

LZW COMPRESSION

- Ziv and Lempel (2002) created a family of loss-less compression algorithms.
- The general scheme consistent among their algorithms is by replacing a repeated sequence of characters by a reference to a previous occurrence.
- LZW was created by Lempel, Ziv and Welch as an improvement to an earlier algorithm, LZ78.
- Algorithm:
  1. Initialize the table to contain all single-byte strings.
  2. Read the first byte, c, from the input buffer. Set the prefix, w, to that byte.
  3. Read the next input byte into c. If at end of buffer, exit.
  4. If wc is in the dictionary then set w to wc and continue reading (step 3).
  5. Store wc in the dictionary, set w to c and continue reading (step 3).
- In short: this process essentially generates frequently occurring n-grams of various lengths.

IMPLEMENTATION

- The language models were built from translated versions of *European Union Human Rights Declaration*, freely available in over fifty languages.
- For each language generate a model – essentially a dictionary derived from LZW compression of training text.
- No pre-processing performed on the training data.
- Dictionary for each model limited to 8k entries.
- Classification: Evaluate an unlabeled document by compressing the document with each model. The model with the highest compression ratio is chosen as the the document language.
- 43 supported languages: Afrikaans, Albanian, Arabic, Armenian, Basque, Bulgarian, Catalan, Chinese (Simplified / Traditional), Czech, Danish, Dutch, English, Estonian, Finnish, French, Georgian, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Korean, Latvian, Lithuanian, Macedonian, Malaysian, Norwegian (Bokmal / Nynorsk), Polish, Portuguese, Romanian, Russian, Slovakian, Spanish, Swedish, Tagalog, Thai, Turkish, Ukrainian, Vietnamese

EVALUATION CORPUS

1. Test set compiled by native speakers of Italian, English, Japanese, Portuguese and Spanish.
2. Search terms were compiled for each language.
3. Using (now deprecated) the Twitter search REST API, tweets were downloaded using those search terms.
4. Finally, each language set of tweets were manually reviewed (limited to 20 tweets). To be considered in a specific language, a tweet had to be composed of at least 80% words in that language.

RESULTS

The following results for 20 tweets per language

Language	Accuracy	Errors
Italian	85%	Spanish
English	80%	Afrikans
Japanese	100%	N/A
Portuguese	60%	Spanish
Spanish	75%	Italian/Portuguese

Incorrect tweets were composed of abbreviations (e. g. 'RT'), foreign loan-words, slang, hashtags (e. g. #nwnlp2014) and URLs. Removing non-linguistic artifacts would most likely increase accuracy especially in small text applications.

CONCLUSIONS

- LZW compression methodology to perform LID on short texts show promise, but more work needed.
- Benefits of using compression techniques are primarily in the ease of implementation and lack of pre-processing during run-time language identification.

FUTURE WORK

- Expand test set size and compare with other techniques.
- Fold case in both training and run-time data.
- Consider the top two languages as opposed to only the top language.
- If tweet contains an URL, fetch and perform language identification of URL content to re-rank or confirm tweet language ranking.