Automatic cognate identification with gap-weighted string subsequences.

Anonymous

Abstract

In this paper, we describe the problem of cognate identification in NLP. We introduce the idea of gap-weighted subsequences for discriminating cognates from non-cognates. We also propose a scheme to integrate phonetic features into the feature vectors for cognate identification. We show that subsequence based features perform better than state-of-the-art classifier for the purpose of cognate identification. The contribution of this paper is the use of subsequence features for cognate identification.

1 Introduction

Cognates are words across languages whose origin can be traced back to a common ancestral language. For example, English $\sim$ German *night* $\sim$ *Nacht* 'night' and; English *hound* $\sim$ German *Hund* 'dog' are cognates whose origin can be traced back to Proto-Germanic. Sometimes, cognates are not revealingly similar but have changed substantially over time such that they do not share form similarity. An example of such a cognate pair is the English *wheel* and Sanskrit *chakra* 'wheel', which can be traced back to Proto-Indo-European (PIE) $*k^wek^welo$.

Automatic cognate identification, in NLP, refers to the application of string similarity or phonetic similarity algorithms either independently, or in tandem with machine learning algorithms for determining if a given word pair is cognate or not (Inkpen et al., 2005). In NLP, even borrowed words (*loanwords*) are referred to as cognates. In contrast, historical linguistics makes a stark distinction between loanwords and cognates. For example, English *beef* is a loanword from Norman French.

In this paper, we use cognates to refer to those words whose origin can be traced back to a common ancestor. We use string subsequence based features (motivated from string kernels) for automatic cognate identification. We show that subsequence based features outperform word similarity measures at the task of automatic cognate identification. We motivate the use of subsequence based features in terms of linguistic examples and then proceed to formulate the subsequence based features that can be derived from string kernels (Shawe-Taylor and Cristianini, 2004). In IR literature, string subsequences go under the name of skip-grams (Järvelin et al., 2007).

2 Related work

The approaches developed by Kondrak and Sherif (2006) and Inkpen et al. (2005) supply different string distances between a pair of words as features to a linear classifier. Usually, a linear classifier such as SVM is trained on labeled positive ("cognates") and negative ("non-cognates") examples and tested on a held-out dataset. Basic vocabulary lists such as the ones devised by Morris Swadesh (Swadesh, 1952), provide a suitable testing ground for applying machine learning algorithms to automatically identify cognates. Some standardized word lists come with cognate information and, subsequently, is used to infer the relationship between languages under purview (Dyen et al., 1992).

Ellison and Kirby (2006) use scaled edit distance (normalized by average length) to measure the intra-lexical divergence in a language. The inter-language distance matrix is supplied to a clustering algorithm to infer a tree for the IE language. The authors only perform a qualitative evaluation of the inferred tree. The authors mention string kernels but do not pursue this line of research further.

Bouchard-Côté et al. (2013) employ a graphical model to reconstruct the proto-word forms from the synchronic word-forms for the Austronesian language family. They compare their automated reconstructions with the ones reconstructed by historical linguists and find that their model beats an edit-distance baseline. However, their model has a strict requirement that the tree structure between the languages under study has to be known before-hand.

Hauer and Kondrak (2011) – referred to as HK – supply different string similarity scores as features to a SVM classifier for determining if a given word pair is a cognate or not. The authors also employ an additional binary language-pair feature – that is used to weigh the language distance – and find that the additional feature assists the task of semantic clustering of cognates. In this task, the cognacy judgments given by a linear classifier is used to flat cluster the lexical items belonging to a single concept. The clustering quality is evaluated against the gold standard cognacy judgments. Unfortunately, the experiments of these scholars cannot be replicated since the partitioning details of their training and test datasets is not available.

In our experiments, we compare our system's performance with the performance of the classifiers trained from HK-based features. In the next section, we will describe string similarity measures, subsequences features, dataset, and results.

3 Cognate identification

3.1 String similarity features: Some issues

Edit distance counts the minimum number of insertions, deletions, and substitutions required to transform one word into another word. Identical words have 0 edit distance. For example, the edit distance between two cognates English *hound* and German *hund* is 1. Similarly, the edit distance between Swedish *i* and Russian *v* 'in', which are cognates, is 1. The edit distance treats both of the cognates at the same level and does not reflect the amount of change which has occurred in the Swedish and Russian words from the PIE word.

Dice is another string similarity measure that defines similarity between two strings as the ratio between the number of common bigrams to the total number of bigrams. The similarity between Lusatian *dolhi* and Czech *dluhe* 'long' is 0 since they do not share any common bigrams and the edit distance between the two strings is 3. Although the two words share all the consonants, the Dice score is 0 due to the intervening vowels.

Another string similarity measure, Longest Common Subsequence (LCS) measures the length of the longest common subsequence between the two words. The LCS is 4 (*hund*), 0 (*i* and *v*), and 3 (*dlh*) for the above examples. One can parade a number of examples which are problematical for the simple-minded string similarity measures. Alternatively, string kernels in machine learning research offer a way to exploit the similarities between two words without any restrictions on the length and character similarity.

3.2 Subsequence features

Subsequences as formulated below weigh the similarity between two words based on the number of dropped characters and combine phoneme classes seamlessly. Having motivated why subsequences seems to be a good idea, we formulate subsequences below.

We follow the notation given in Shawe-Taylor and Cristianini (2004) to formulate our representation of a string (word). Let Σ denote the set of phonetic alphabet. Given a string s over Σ, the subsequence vector $\mathbf{\Phi(s)}$ is defined as follows. The string s can decomposed as $s_1, \ldots, s_{|s|}$ where $|s|$ denotes the length of the string. Let $\overrightarrow{I}$ denote a sequence of indices $(i_1, \ldots, i_{|u|})$ where, $1 \leq i_1 < \ldots < i_{|u|} \leq |s|$. Then, a subsequence u is a sequence of characters $s[\overrightarrow{I}]$. Note that a subsequence can occur multiple times in a string. Then, the weight of u, $\phi_u(s)$ is defined as $\sum_{\overrightarrow{I}:u=s[\overrightarrow{I}]} \lambda^{l(\overrightarrow{I})}$ where, $l(\overrightarrow{I}) = i_{|u|} - i_1 + 1$ and $\lambda \in (0, 1)$ is a decay factor.

The subsequence vector $\Phi(s)$ is composed of $< \phi_u(s) > \forall u \in \bigcup_{n=1}^{p} \Sigma^n$ where $1 \leq n \leq p$ is the length of u and p is the maximum length of the subsequences. As $\lambda \to 0$, a subsequence is constrained to a substring. As $\lambda \to 1$, $\phi_u(s)$ counts the frequency of u in s. We also experiment with different values of λ in our experiments.

The λ factor is exponential and penalizes u over long gaps in a string. Due to the above formulation, the frequency of a subsequence u in a single string is also taken into account. The subsequence formulation also allows for the incorporation of a class-based features easily. For instance, each σ in u can be mapped to its **C**onsonant/**V**owel class: $\sigma \mapsto \{C, V\}$. The subsequence formulation also allows us to map each phonetic symbol (for example, from International Phonetic Alphabet [IPA]) to an intermediary phonetic alphabet also. Unfortunately, the current dataset is not transcribed in IPA to convert it into an intermediary broader format. In this paper, we map each string s into its C, V sequence s_{cv} and then compute the subsequence weights.[1]

A combined subsequence vector $\Phi(s+s_{cv})$ is further normalized by its norm, $\|\Phi(s+s_{cv})\|$, to transform into a unit vector. The common subsequence vector $\Phi(s_1, s_2)$ is composed of all the common subsequences between s_1 and s_2. The weight of a common subsequence is $\phi_u(s_1) + \phi_u(s_2)$.

Moschitti et al. (2012) list the features of the above weighting scheme.

- Longer subsequences receive lower weights.
- Characters can be omitted (called *gaps*).
- The exponent of λ penalizes recurring subsequences with longer gaps.

For a string of length m and subsequence length n, the computational complexity is in the order of $\mathcal{O}(mn)$.

On a linguistic note, gaps are consistent with the prevalent sound changes such as sound loss, sound gain, and assimilation[2], processes which alter word forms in an ancestral language causing the daughter languages to have different surface forms. The λ factor weighs the number of gaps found in a subsequence. For instance, the Sardinian word form for 'fish' *pissi* has the subsequence *ps* occurring twice but with different weights: λ^3, λ^4. Hence, *ps*'s weight is $\lambda^3 + \lambda^4$. On another note, the idea of gap subsequences subsumes the definitions of different n-gram similarities introduced by Kondrak (2005).

The combined feature vector, for a word pair, is used to train a SVM classifier. In our experiments, we use the LIBLINEAR package (Fan et al., 2008) to solve the primal problem with L_2-regularization and L_2-loss. The next subsection describes the makeup of the dataset. We use the default SVM parameters since we did not observe any difference in our development experiments.

3.3 Dataset and results

In this section, we will present the dataset, HK features, and results of our experiments.

Dataset. We used the publicly available[3] Indo-European dataset (Dyen et al., 1992) for our experiments. The dataset has $16,520$ lexical items for 200 concepts and 84 language varieties. Each word form is assigned to a unique CCN (Cognate Class Number). There are more than 200 identical non-cognate pairs in the dataset. For the first experiment, we extracted all word pairs for a concept and assigned a positive label if the word pair has an identical CCN; a negative label, if the word pair has different CCNs. We extracted a total of $619,635$ word pairs out of which $145,145$ are cognates. The dataset is transcribed in a broad romanized phonetic alphabet.

We explored if we could use two other word list databases: ASJP (Brown et al., 2008) and Ringe et al. (2002) for our experiments. Although the ASJP database has word lists for more than half of the world's languages, it has cognacy judgments for few selected languages and is limited to 40 concepts. Moreover, the ASJP database does not have cognacy judgments for Indo-European family. The other dataset of Ringe et al. (2002) has items for 17 ancient Indo-European languages which are transcribed in an orthographic format and not in a phonetic alphabet. Moreover, there are a number of missing items in this dataset. Hence, we did not use Ringe et al.'s dataset in our experiments. In contrast, Dyen's dataset is much larger and transcribed in an uniform format. Now, we proceed to describe the previous best-performing system.

HK's system. We compare the performance of subsequence features against the SVM classifier system trained on the following word-similarity fea-

[1] $V = \{a, e, i, o, u, y\}, C = \Sigma \setminus V$.

[2] A sound can assimilate to a neighboring sound. Sanskrit *agni* > Prakrit *aggi* 'fire'. Compare the Latin form *ignis*.

[3] http://www.wordgumbo.com/ie/cmp/iedata.txt

tures from Hauer and Kondrak (2011):

- Edit distance.
- Length of longest common prefix.
- Number of common bigrams.
- Lengths of individual words.
- Absolute difference between the lengths of the words.

Cross-Validation experiment. As a first step, we perform a random ten-fold cross-validation of the dataset and report the accuracies for various values of λ and p. The results of this experiment are shown in figure 1. The best results are obtained at $\lambda = 0.8, p = 3$. The accuracies increase with an increment in the value of λ until 0.8 for all $p > 1$ (non-unigram models). This experiment is mainly designed to test the robustness of subsequence features against random splits in the dataset which is shown to be robust. The subsequence features outperform HK-based classifier in this experiment.

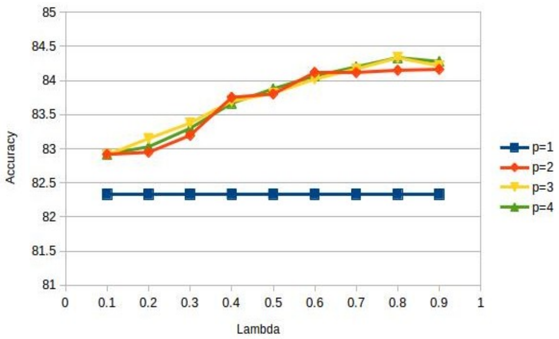

Figure 1: Ten-fold cross-validation accuracy for incremental λ and p. HK system's accuracy is 82.61%.

	positive	negative
training	$111,918$	$353,957$
test	$33,227$	$120,533$

Table 1: Number of positive and negative examples in the training and test sets. The ratio of positive to negative examples is $1 : 3.62$.

Concepts experiment. In this experiment, we split our dataset into two sets of concepts; and train on one set and test on the other. To replicate our dataset, we performed an alphabetical sort of the concepts and split the concepts into training and testing datasets with a ratio of $3 : 1$. Now, we extract positive and negative examples from each subset of concepts; and train and test on each concepts' subset. We also performed a 3-fold cross-validation on the training set to tune c (SVM hyperparameter). We observed that the value of c did not effect the cross-validation accuracy on the training dataset. Hence we fixed c at 1. We also experimented with radial-basis function kernel and polynomial kernels but did not find any improvement over the linear kernel classifier. The composition of the training and test sets is given in table 1.

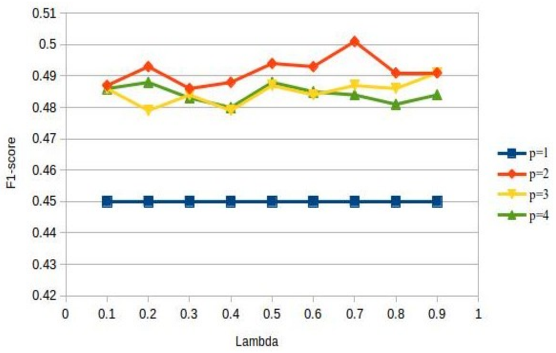

Figure 2: F_1-score for different values of p and λ. HK system's F_1-score is 0.46.

In this experiment, we report the F_1-score, defined as $\frac{2PR}{P+R}$ (**P**recision and **R**ecall), for different values of λ and p. The results of this experiment are shown in figure 2. HK system's F_1-score is 0.46 whereas the best performing subsequence system ($\lambda = 0.7, p = 2$) has a score of 0.5. Our system performs better than the HK system in terms of cross-validation accuracy as well as F_1-score. Overall, all non-unigram models perform better than HK system at cross-validation and concepts experiments.

4 Conclusion

In this paper, we proposed a string kernel based approach for the purpose of cognate identification. We formulated an approach to integrate phonetic features of a phonetic symbol into the feature vector and showed that it beats HK system at cognate identification at cross-validation and concepts subsets.

In future, we plan to make a larger dataset of cognacy judgments for other language families in a richer phonetic transcription and integrate articulatory phonetic features into the feature vectors for the purpose of cognate identification. We also plan on testing with different feature vector combinations.

References

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 110(11):4224–4229.

Cecil H. Brown, Eric W. Holman, Søren Wichmann, and Viveka Velupillai. 2008. Automated classification of the world's languages: A description of the method and preliminary results. *Sprachtypologie und Universalienforschung*, 61(4):285–308.

Isidore Dyen, Joseph B. Kruskal, and Paul Black. 1992. An Indo-European classification: A lexicostatistical experiment. *Transactions of the American Philosophical Society*, 82(5):1–132.

T. Mark Ellison and Simon Kirby. 2006. Measuring language divergence by intra-lexical comparison. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 273–280, Sydney, Australia, July. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Bradley Hauer and Grzegorz Kondrak. 2011. Clustering semantically equivalent words into cognate sets in multilingual lists. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 865–873, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Diana Inkpen, Oana Frunza, and Grzegorz Kondrak. 2005. Automatic identification of cognates and false friends in French and English. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 251–257.

Anni Järvelin, Antti Järvelin, and Kalervo Järvelin. 2007. s-grams: Defining generalized n-grams for information retrieval. *Information Processing & Management*, 43(4):1005–1019.

Grzegorz Kondrak and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of ACL Workshop on Linguistic Distances*, pages 43–50. Association for Computational Linguistics.

Grzegorz Kondrak. 2005. N-gram similarity and distance. In *String Processing and Information Retrieval*, pages 115–126. Springer.

Alessandro Moschitti, Qi Ju, and Richard Johansson. 2012. Modeling topic dependencies in hierarchical text categorization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 759–767. Association for Computational Linguistics.

Don Ringe, Tandy Warnow, and Ann Taylor. 2002. Indo-European and computational cladistics. *Transactions of the Philological Society*, 100(1):59–129.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel methods for pattern analysis*. Cambridge university press.

Morris Swadesh. 1952. Lexico-statistic dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos. *Proceedings of the American philosophical society*, 96(4):452–463.