Evaluation of Several Phonetic Similarity Algorithms on the Task of Cognate Identification

Grzegorz Kondrak and **Tarek Sherif**
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
{kondrak,tarek}@cs.ualberta.ca

Abstract

We investigate the problem of measuring phonetic similarity, focusing on the identification of cognates, words of the same origin in different languages. We compare representatives of two principal approaches to computing phonetic similarity: manually-designed metrics, and learning algorithms. In particular, we consider a stochastic transducer, a Pair HMM, several DBN models, and two constructed schemes. We test those approaches on the task of identifying cognates among Indoeuropean languages, both in the supervised and unsupervised context. Our results suggest that the averaged context DBN model and the Pair HMM achieve the highest accuracy given a large training set of positive examples.

1 Introduction

The problem of measuring phonetic similarity between words arises in various contexts, including speech processing, spelling correction, commercial trademarks, dialectometry, and cross-language information retrieval (Kessler, 2005). A number of different schemes for computing word similarity have been proposed. Most of those methods are derived from the notion of edit distance. In its simplest form, edit distance is the minimum number of edit operations required to transform one word into the other. The set of edit operations typically includes substitutions, insertions, and deletions, and may incorporate more complex transformations.

By assigning variable weights to various edit operations depending on the characters involved in the operations, one can design similarity schemes that are more sensitive to a given task. Such variable weight schemes can be divided into two main groups. One approach is to manually design edit operation weights on the basis of linguistic intuition and/or physical measurements. Another approach is to use machine learning techniques to derive the weights automatically from training data composed of a set of word pairs that are considered similar. The manually-designed schemes tend to be somewhat arbitrary, but can be readily applied to diverse tasks. The learning approaches are also easily adaptable to various tasks, but they crucially require training data sets of reasonable size. In general, the more complex the underlying model, the larger the data sets needed for parameter estimation.

In this paper, we focus on a few representatives of both approaches, and compare their performance on the specific task of cognate identification. Cognate identification is a problem of finding, in distinct languages, words that can be traced back to a common word in a proto-language. Beyond historical linguistics, cognate identification has applications in other areas of computational linguistics (Mackay and Kondrak, 2005). Because the likelihood that two words across different languages are cognates is highly correlated with their phonetic similarity, cognate identification provides an objective test of the quality of phonetic similarity schemes.

The remainder of this paper is organized as fol-

lows. Section 2 discusses the two manually designed schemes: the ALINE algorithm and a linguistically-motivated metric. Section 3 discusses various learning approaches. In Section 4, we describe Dynamic Bayesian Nets. Finally, in Section 5, we discuss the results of our experiments.

2 Two manually constructed schemes

In this section, we first describe two different constructed schemes and then compare their properties.

2.1 ALINE

The ALINE algorithm (Kondrak, 2000) assigns a similarity score to pairs of phonetically-transcribed words on the basis of the decomposition of phonemes into elementary phonetic features. The algorithm was originally designed to identify and align cognates in vocabularies of related languages. Nevertheless, thanks to its grounding in universal phonetic principles, the algorithm can be used for estimating the similarity of any pair of words.

The principal component of ALINE is a function that calculates the similarity of two phonemes that are expressed in terms of about a dozen multi-valued phonetic features (*Place*, *Manner*, *Voice*, etc.). The phonetic features are assigned *salience* weights that express their relative importance. Feature values are encoded as floating-point numbers in the range $[0, 1]$. For example, the feature *Manner* can take any of the following seven values: *stop* = 1.0, *affricate* = 0.9, *fricative* = 0.8, *approximant* = 0.6, *high vowel* = 0.4, *mid vowel* = 0.2, and *low vowel* = 0.0. The numerical values reflect the distances between vocal organs during speech production.

The overall similarity score is the sum of individual similarity scores between pairs of phonemes in an optimal alignment of two words, which is computed by a dynamic programming algorithm (Wagner and Fischer, 1974). A constant insertion/deletion penalty is applied for each unaligned phoneme. Another constant penalty is set to reduce relative importance of vowel—as opposed to consonant—phoneme matches. The similarity value is normalized by the length of the longer word.

ALINE's behavior is controlled by a number of parameters: the maximum phonemic score, the insertion/deletion penalty, the vowel penalty, and the feature salience weights. The parameters have default settings for the cognate matching task, but these settings can be optimized (tuned) on a development set that includes both positive and negative examples of similar words.

2.2 A linguistically-motivated metric

Phonetically natural classes such as /p b m/ are much more common among world's languages than unnatural classes such as /o z g/. In order to show that the bias towards phonetically natural patterns of phonological classes can be modeled without stipulating phonological features, Mielke (2005) developed a phonetic distance metric based on acoustic and articulatory measures. Mielke's metric encompasses 63 phonetic segments that are found in the inventories of multiple languages. Each phonetic segment is represented by a 7-dimensional vector that contains three acoustic dimensions and four articulatory dimensions (perceptual dimensions were left out because of the difficulties involved in comparing almost two thousand different sound pairs). The phonetic distance between any two phonetic segments were then computed as the Euclidean distance between the corresponding vectors.

For determining the acoustic vectors, the recordings of 63 sounds were first transformed into waveform matrices. Next, distances between pairs of matrices were calculated using the Dynamic Time Warping technique. These acoustic distances were subsequently mapped to three acoustic dimensions using multidimensional scaling. The three dimensions can be interpreted roughly as (a) sonorous vs. sibilant, (b) grave vs. acute, and (c) low vs. high formant density.

The articulatory dimensions were based on ultrasound images of the tongue and palate, video images of the face, and oral and nasal airflow measurements. The four articulatory dimensions were: oral constriction location, oral constriction size, lip constriction size, and nasal/oral airflow ratio.

2.3 Comparison

When ALINE was initially designed, there did not exist any concrete linguistically-motivated similarity scheme to which it could be compared. Therefore, it is interesting to perform such a comparison with the recently proposed metric.

The principal difficulty in employing the metric for computing word similarity is the limited size of the phonetic segment set, which was dictated by practical considerations. The underlying database of phonological inventories representing 610 languages contains more than 900 distinct phonetic segments, of which almost half occur in only one language. However, because a number of complex measurements have to be performed for each sound, only 63 phonetic segments were analyzed, which is a set large enough to cover only about 20% of languages in the database. The set does not include such common phones as dental fricatives (which occur in English and Spanish), and front rounded vowels (which occur in French and German). It is not at all clear how one to derive pairwise distances involving sounds that are not in the set.

In contrast, ALINE produces a similarity score for any two phonetic segment so long as they can be expressed using the program's set of phonetic features. The feature set can in turn be easily extended to include additional phonetic features required for expressing unusual sounds. In practice, any IPA symbol can be encoded as a vector of universal phonetic features.

Another criticism that could be raised against Mielke's metric is that it has no obvious reference point. The choice of the particular suite of acoustic and articulatory measurements that underlie the metric is not explicitly justified. It is not obvious how one would decide between different metrics for modeling phonetic generalizations if more than one were available.

On the other hand, ALINE was designed with a specific reference in mind, namely cognate identification. The "goodness" of alternative similarity schemes can be objectively measured on a test set containing both cognates and unrelated pairs from various languages.

A perusal of individual distances in Mielke's metric reveals that some of them seem quite unintuitive. For example, [t] is closer to [j] than it is to [ʦ], [ə] is closer to [n] than to [i], [ʒ] is closer to [e] than to [g]. etc. This may be caused either by the omission of perceptual features from the underlying set of features, or by the assignment of uniform weights to different features (Mielke, *personal communication*).

It is difficult to objectively measure which phonetic similarity scheme produces more "intuitive" values. In order to approximate a human evaluation, we performed a comparison with the perceptual judgments of Laver (1994), who assigned numerical values to pairwise comparisons of 22 English consonantal phonemes on the basis of "subjective auditory impressions". We counted the number of perceptual conflicts with respect to Laver's judgments for both Mielke's metric and ALINE's similarity values. For example, the triple ([ʃ], [j], [k]) is an example of a conflict because [ʃ] is considered closer to [j] than to [k] in Mielke's matrix but the order is the opposite in Laver's matrix. The program identified 1246 conflicts with Mielke's metric, compared to 1058 conflicts with ALINE's scheme, out of 4620 triples. We conclude that in spite of the fact that ALINE is designed for identifying cognates, rather than directly for phonetic similarity, it is more in agreement with human perceptual judgments than Mielke's metric which was explicitly designed for quantifying phonetic similarity.

3 Learning algorithms

In this section, we briefly describe several machine learning algorithms that automatically derive weights or probabilities for different edit operations.

3.1 Stochastic transducer

Ristad and Yianilos (1998) attempt to model edit distance more robustly by using Expectation Maximization to learn probabilities for each of the possible edit operations. These probabilities are then used to create a stochastic transducer, which scores a pair of words based on either the most probable sequence of operations that could produce the two words (Viterbi scoring), or the sum of the scores of all possible paths that could have produced the two words (stochastic scoring). The score of an individual path here is simply the product of the probabilities of the edit operations in the path. The algorithm was evaluated on the task of matching surface pronunciations in the Switchboard data to their canonical pronunciations in a lexicon, yielding a significant improvement in accuracy over Levenshtein distance.

3.2 Levenshtein with learned weights

Mann and Yarowsky (2001) applied the stochastic transducer of Ristad and Yianilos (1998) for inducing translation lexicons between two languages, but found that in some cases it offered no improvement over Levenshtein distance. In order to remedy this problem, they they proposed to filter the probabilities learned by EM into a few discrete cost classes, which are then used in the standard edit distance algorithm. The LLW approach yielded improvement over both regular Levenshtein and the stochastic transducer.

3.3 CORDI

CORDI (Kondrak, 2002) is a program for detecting recurrent sound correspondences in bilingual wordlists. The idea is to relate recurrent sound correspondences in wordlists to translational equivalences in bitexts. A *translation model* is induced between phonemes in two wordlists by combining the maximum similarity alignment with the competitive linking algorithm of Melamed (2000). Melamed's approach is based on the *one-to-one* assumption, which implies that every word in the bitext is aligned with at most one word on the other side of the bitext. In the context of the bilingual wordlists, the correspondences determined under the *one-to-one* assumption are restricted to link single phonemes to single phonemes. Nevertheless, the method is powerful enough to determine valid correspondences in wordlists in which the fraction of cognate pairs is well below 50%.

The discovered phoneme correspondences can be used to compute a correspondence-based similarity score between two words. Each valid correspondence is counted as a link and contributes a constant positive score (no crossing links are allowed). Each unlinked segment, with the exception of the segments beyond the rightmost link, is assigned a smaller negative score. The alignment with the highest score is found using dynamic programming (Wagner and Fischer, 1974). If more than one best alignment exists, links are assigned the weight averaged over the entire set of best alignments. Finally, the score is normalized by dividing it by the average of the lengths of the two words.

3.4 Pair HMM

Mackay and Kondrak (2005) propose to computing similarity between pairs of words with a technique adapted from the field of bioinformatics. A Pair Hidden Markov Model differs form a standard HMM by producing two output streams in parallel, each corresponding to a word that is being aligned. The model has three states that correspond to the basic edit operations: substitution, insertion, and deletion. The parameters of the model are automatically learned from training data that consists of word pairs that are known to be similar. The model is trained using the Baum-Welch algorithm (Baum et al., 1970).

4 Dynamic Bayesian Nets

A Bayesian Net is a directed acyclic graph in which each of the nodes represents a random variable. The random variable can be either deterministic, in which case the node can only take on one value for a given configuration of its parents, or stochastic, in which case the configuration of the parents determines the probability distribution of the node. Arcs in the net represent dependency relationships.

Filali and Bilmes (2005) proposed to use Dynamic Bayesian Nets (DBNs) for computing word similarity. A DBN is a Bayesian Net where a set of arcs and nodes are maintained for each point in time in a dynamic process. This involves set of prologue frames denoting the beginning of the process, chunk frames which are repeated for the middle of the process, and epilogue frames to end the process. The conditional probability relationships are time-independent. DBNs can encode quite complex interdependencies between states.

We tested four different DBN models on the task of cognate identification. In the following description of the models, Z denotes the current edit operation, which can be either a substitution, an insertion, or a deletion.

MCI The *memoriless context-independent model* (Figure 1) is the most basic model, which is meant to be equivalent to the stochastic transducer of Ristad and Yianilos (1998). Its lack of memory signifies that the probability of Z taking on a given value does not depend in any way on what previous values of Z have been. The context-independence refers to the fact that

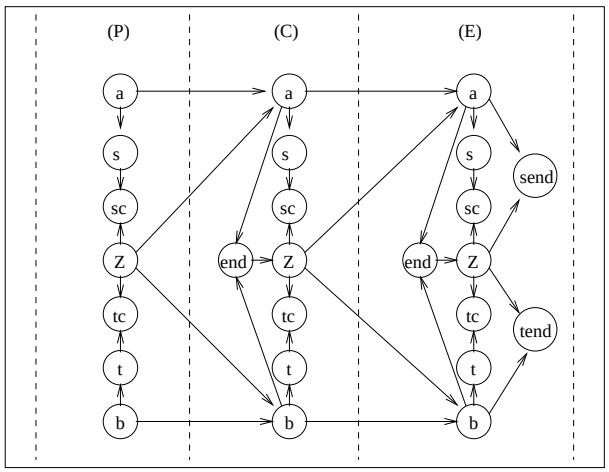

Figure 1: The MCI model.

the probability of Z taking on a certain value does not depend on the letters of the source or target word. The a and b nodes in Figure 1 represent the current position in the source and target words, respectively. The s and t nodes represent the current letter in the source and target words. The *end* node is a switching parent of Z and is triggered when the values of the a and b nodes move past the end of both the source and target words. The sc and tc nodes are consistency nodes which ensure that the current edit operation is consistent with the current letters in the source and target words. Consistency here means that the source side of the edit operation must either match the current source letter or be ε, and that the same be true for the target side. Finally, the *send* and *tend* nodes appear only in the last frame of the model, and are only given a positive probability if both words have already been completely processed, or if the final edit operation will conclude both words. The following models all use the MCI model as a basic framework, while adding new dependencies to Z.

MEM In the *memory model*, the probability of the current operation being performed depends on what the previous operation was.

CON In the *context-dependent model*, the probability that Z takes on certain values is dependent on letters in the source word or target word. The model that we test in Section 5, takes into account the context of two letters in the source word: the current one and the immediately preceding one. We experimented with several other variations of context sets, but they either performed poorly on the development set, or required inordinate amounts of memory.

LEN The *length model* learns the probability distribution of the number of edit operations to be performed, which is the incorporated into the similarity score. This model represents an attempt to counterbalance the effect of longer words being assigned lower probabilities.

The models were implemented with the GMTK toolkit (Bilmes and Zweig, 2002). A more detailed description of the models can be found in (Filali and Bilmes, 2005).

5 Experiments

5.1 Setup

We evaluated various methods for computing word similarity on the task of the identification of cognates. The input consists of pairs of words that have the same meaning in distinct languages. For each pair, the system produces a score representing the likelihood that the words are cognate. Ideally, the scores for true cognate pairs should always be higher than scores assigned to unrelated pairs. For binary classification, a specific score threshold could be applied, but we defer the decision on the precision-recall trade-off to downstream applications. Instead, we order the candidate pairs by their scores, and evaluate the ranking using *11-point interpolated average precision* (Manning and Schutze, 2001). Scores are normalized by the length of the longer word in the pair.

Word similarity is not always a perfect indicator of cognation because it can also result from lexical borrowing and random chance. It is also possible that two words are cognates and yet exhibit little surface similarity. Therefore, the upper bound for average precision is likely to be substantially lower than 100%.

Languages		Proportion of cognates	Method						
			EDIT	MIEL	ALINE	R&Y	LLW	PHMM	DBN
English	German	0.590	0.906	0.909	0.912	0.894	0.918	0.930	0.927
French	Latin	0.560	0.828	0.819	0.862	0.889	0.922	0.934	0.923
English	Latin	0.290	0.619	0.664	0.732	0.728	0.725	0.803	0.822
German	Latin	0.290	0.558	0.623	0.705	0.642	0.645	0.730	0.772
English	French	0.275	0.624	0.623	0.623	0.684	0.720	0.812	0.802
French	German	0.245	0.501	0.510	0.534	0.475	0.569	0.734	0.645
Albanian	Latin	0.195	0.597	0.617	0.630	0.568	0.602	0.680	0.676
Albanian	French	0.165	0.643	0.575	0.610	0.446	0.545	0.653	0.658
Albanian	German	0.125	0.298	0.340	0.369	0.376	0.345	0.379	0.420
Albanian	English	0.100	0.184	0.287	0.302	0.312	0.378	0.382	0.446
AVERAGE		0.2835	**0.576**	**0.597**	**0.628**	**0.601**	**0.637**	**0.704**	**0.709**

Table 1: 11-point average cognate identification precision for various methods.

5.2 Data

The training data for our cognate identification experiments comes from the Comparative Indoeuropean Data Corpus (Dyen et al., 1992). The data contains word lists of 200 basic meanings representing 95 speech varieties from the Indoeuropean family of languages. Each word is represented in an orthographic form without diacritics using the 26 letters of the Roman alphabet. Approximately 180,000 cognate pairs were extracted from the corpus.

The development set was composed of three language pairs: Italian-Croatian, Spanish-Romanian, and Polish-Russian. We chose these three language pairs because they represent very different levels of relatedness: 25.3%, 58.5%, and 73.5% of the word pairs are cognates, respectively. The percentage of cognates within the data is important, as it provides a simple baseline from which to compare the success of our algorithms. If our cognate identification process were random, we would expect to get roughly these percentages for our recognition precision (on average).

The test set consisted of five 200-word lists representing English, German, French, Latin, and Albanian, compiled by Kessler (2001). The lists for these languages were removed from the training data (except Latin, which was not part of the training set), in order to keep the testing and training data as separate as possible. For the supervised experiments, we converted the test data to have the same orthographic representation as the training data.

The training process for the DBN models consisted of three iterations of Expectation Maximization, which was determined to be optimal on the development data. Each pair was used twice, once in each source-target direction, to enforce the symmetry of the scoring, One of the models, the context-dependent model, remained asymmetrical despite to two-way training. In order to remove the undesirable asymmetry, we averaged the scores in both directions for each word pair.

5.3 Results

Table 1 shows the average cognate identification precision on the test set for a number of methods. EDIT is a baseline edit distance with uniform costs. MIEL refers to edit distance with weights computed using the approach outlined in (Mielke, 2005). ALINE denotes the algorithm for aligning phonetic sequences (Kondrak, 2000) described in Section 2.1. R&Y is the stochastic transducer of Ristad and Yianilos (1998). LLW stands for *Levenshtein with learned weights*, which is a modification of R&Y proposed by Mann and Yarowsky (2001). The PHMM column provides the results reported in (Mackay and Kondrak, 2005) for the best Pair HMM model, which uses log odds scoring. Finally, DBN stands for our best results obtained with a DBN model, in this case the averaged context model.

Table 2 show the aggregate results for various DBN models. Two different results are given for each model: the raw score, and the score normal-

Model	Raw Score	Normalized
MCI	0.515	0.601
MEM	0.563	0.595
LEN	0.516	0.587
CON-FOR	0.582	0.599
CON-REV	0.624	0.619
CON-AVE	0.629	0.709

Table 2: Average cognate identification precision for various DBN models.

Model	Raw Score	Normalized
MCI	0.462	0.430
MEM	0.351	0.308
LEN	0.464	0.395
CON-AVE	0.433	0.414
CORDI	—	0.629

Table 3: Phonetic test results.

ized by the length of the longer word. The models are the memoriless context-independent model (MCI), memory model (MEM), length model (LEN) and context model (CON). The context model results are split as follows: results in the original direction (FOR), results with all word pairs reversed (REV), and the results of averaging the scores for each word pair in the forward and reverse directions (AVE).

Table 3 shows the aggregate results for the unsupervised approaches. In the unsupervised tests, the training set was not used, as the models were trained directly on the testing data without access to the cognation information. For the unsupervised tests, the original, the test set was in its original phonetic form. The table compares the results obtained with various DBN models and with the CORDI algorithm described in Section 3.3.

5.4 Discussion

The results in Table 1 strongly suggest that the learning approaches are more effective than the manually-designed schemes for cognate identification. However, it has to be remembered that the learning process was conducted on a relatively large set of Indoeuropean cognates. Even though there was no overlap between the training and the test set, the latter also contained cognate pairs from the same language family. For each of the removed languages, there are other closely related languages that are retained in the training set, which may exhibit similar or even identical regular correspondences.

The manually-designed schemes have the advantage of not requiring any training sets after they have been developed. Nevertheless, Mielke's metric appears to produce only small improvement over simple edit distance. ALINE outperforms Mielke's metric, which is not surprising considering that ALINE was developed specifically for identifying cognates, and Mielke's substitution matrix lacks several phonemes that occur in the test set.

Among the DBN models, the average context model performs the best. The averaged context model is clearly better than either of the unidirectional models on which it is based. It is likely that the averaging allows the scoring to take contextual information from both words into account, instead of just one or the other. The averaged context DBN model performs about as well as on average as the Pair HMM approach, but substantially better than the R&Y approach and its modification, LLW.

In the unsupervised context, all DBN models fail to perform meaningfully, regardless of whether the scores are normalized or not. In view of this, it is remarkable that CORDI achieves a respectable performance just by utilizing discovered correspondences, having no knowledge of phonetics nor identity of phonemes. The precision of CORDI is at the same level as the phonetically-based ALINE. In fact, a method that combines ALINE and CORDI achieves the average precision of 0.681 on the same test set (Kondrak, *in preparation*).

In comparison with the results of Filali and Bilmes (2005), certain differences are apparent. The memory and length models, which performed better than the memoriless context-independent model on the pronunciation task, perform worse overall here. This is especially notable in the case of the length model which was the best overall performer on their task. The context-dependent model, however, performed well on both tasks.

As mentioned in (Mann and Yarowsky, 2001), it appears that there are significant differences between the pronunciation task and the cognate iden-

tification task. They offer some hypotheses as to why this may be the case, such as noise in the data and the size of the training sets, but these issues are not apparent in the task presented here. The training set was quite large and consisted only of known cognates. The two tasks are inherently different, in that scoring in the pronunciation task involves finding the best match of a surface pronunciation with pronunciations in a lexicon, while the cognate task involves the ordering of scores relative to each other. Certain issues, such as length of words, may become more prominent in this setup. We countered this by normalizing all scores, which was not done in (Filali and Bilmes, 2005). As can be seen in Table 2, the normalization by length appears to improve the results on average. It notable that normalization even helps the length model on this task, despite the fact that it was designed to take word length into account.

6 Conclusion

We have compared the effectiveness of a number of different methods, including the DBN models, on the task of cognate identification. The results suggest that some of the learning methods, namely the Pair HMMs and the averaged context DBN model, outperform the manually designed methods, provided that large training sets are available.

In the future, we would like to apply DBNs to other tasks involving computing word similarity and/or alignment. An interesting next step would be to use them for tasks involving generation, for example the task of machine transliteration.

Acknowledgments

We would like to thank Karim Filali for the DBN scripts, and for advice about how to use them. Thanks to Jeff Mielke for making his phoneme similarity matrix available for our experiments, and for commenting on the results. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic function of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Jeff Bilmes and Geoffrey Zweig. 2002. The graphical models toolkit: An open source software system for speech and time-series processing. In *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing.*

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

Karim Filali and Jeff Bilmes. 2005. A dynamic Bayesian framework to model context and memory in edit distance learning: An application to pronunciation classification. In *Proceedings of ACL 2005*, pages 338–345.

Brett Kessler. 2001. *The Significance of Word Lists*. Stanford: CSLI Publications, Stanford, California.

Brett Kessler. 2005. Phonetic comparison algorithms. *Transactions of the Philological Society*, 103(2):243–260.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of NAACL 2000*, pages 288–295.

Grzegorz Kondrak. 2002. Determining recurrent sound correspondences by inducing translation models. In *Proceedings of COLING 2002*, pages 488–494.

John Laver. 1994. *Principles of Phonetics*. Cambridge University Press.

Wesley Mackay and Grzegorz Kondrak. 2005. Computing word similarity and identifying cognates with Pair Hidden Markov Models. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*, pages 40–47.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL 2001*, pages 151–158.

Christopher D. Manning and Hinrich Schutze. 2001. *Foundations of Statistical Natural Language Processing*. The MIT Press.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Jeff Mielke. 2005. Modeling distinctive feature emergence. In *Proceedings of the 24th West Cost Conference on Formal Linguistics*, pages 281–289.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173.