Using Machine Learning for Translation Inference Across Dictionaries

Kathrin Donandt, Christian Chiarcos, Maxim Ionov

Goethe-Universität Frankfurt, Germany
{donandt|chiarcos|ionov}@cs.uni-frankfurt.de

Abstract. This paper describes our contribution to the closed track of the Shared Task *Translation Inference across Dictionaries* (TIAD-2017),[1] held in conjunction with the first Conference on Language Data and Knowledge (LDK-2017). In our approach, we use supervised machine learning to predict high-quality candidate translation pairs. We train a Support Vector Machine using several features, mostly of the translation graph, but also taking into consideration string similarity (Levenshtein distance). As the closed track does not provide manual training data, we define positive training examples as translation candidate pairs which occur in a cycle in which there is a direct connection.

1 Introduction

The Shared Task on *Translation Inference across Dictionaries* (TIAD-2017), held in conjunction with the first Conference on Language Data and Knowledge (LDK-2017), aims to develop and test methods and techniques for auto-generating new bilingual dictionaries based on existing resources: Provided with bilingual dictionary fragments for various languages, the task consists in auto-generating bilingual dictionaries. For example, a dictionary for languages A and C is extrapolated from dictionaries for a pivot language B (i.e., $A \mapsto B$ and $B \mapsto C$), or a chain of pivot languages $B, ..., X$. However, natural language is polysemous and a candidate pair $w_A \rightarrow_? w_C$ is thus not guaranteed to be a valid translation if a translation path, say, $w_A \rightarrow_{A \mapsto B} w_B$ and $w_B \rightarrow_{B \mapsto C} w_C$, does exist.

Therefore, we assess the likelihood of inferred candidate pairs using a Support Vector Machine (SVM) to determine correct translation pairs. We participate in the closed track, where only the lexicographic data from multilingual learner dictionaries is to be taken into consideration. Shared Task evaluation is done manually on the output provided by the participating systems. By design, this means that no internal evaluation or training of the system against gold data with the same criteria is possible. In order to pursue our machine learning approach, we synthesize training data from cycles and paths in a translation graph: We define positive training examples as translation candidate pairs which occur in a cycle in which there is a direct connection from a word in the target language

[1] https://tiad2017.wordpress.com/

back to a word in the source language. This means that data originally provided 'to close the loop' contributes to the training data, an approach approved in coordination with the Shared Task organizers.[2]

2 Data and preprocessing

The data used in the Shared Task is provided by KDictionaries Ltd., and consists of excerpts of bilingual learner dictionaries. Dictionary fragments for the following language pairs are provided:

- German (de) ↦ Danish (da), Dutch (nl), English (en), Japanese (jp)
- Danish (da) ↦ French (fr)
- Dutch (nl) ↦ Spanish (es)
- French (fr) ↦ Spanish (es), Brazilian Portuguese (pt-BR)
- Japanese (jp) ↦ Spanish (es)
- Spanish (es) ↦ Brazilian Portuguese (pt-BR), Danish (da)
- English (en) ↦ Brazilian Portuguese (pt-BR)

The dictionary information is not exhaustive, but limited to sample data accounting for a selection of German and Brazilian Portuguese words along the following paths:

- de ↦ en ↦ pt-BR (↦ de)[3]
- de ↦ jp ↦ es ↦ pt-BR (↦ de)
- de ↦ da ↦ fr ↦ es ↦ pt-BR (↦ de)
- de ↦ nl ↦ es ↦ da ↦ fr ↦ pt-BR (↦ de)

The task is to produce dictionaries for three novel language combinations: de ↦ pt-BR, da ↦ es, and nl ↦ fr.

Following the baseline system,[4] we represent dictionary entries in a graph:

- The original bilingual dictionaries are given in a tabular format, with one row comprising seven attributes (word, part-of-speech and example phrase for source word and target word, respectively, as well as an ID, containing the source and target language abbreviation).
- For every source word and every target word, we create a node which contains the word itself word, its language and its part-of-speech.

[2] 'Closing the loop' is the core strategy of the provided baseline system, meaning that translation candidates $w_A \rightarrow_? w_C$ are pre-filtered to instances where a back-translation $w_C \rightarrow_? w_A$ can be extrapolated from the data. In our experiments, we found that using this back-translation information outperforms any approach operating on features of the translation path from w_A to w_C alone.

[3] The Portuguese-German sets are provided for the sole purpose of 'closing the loop', i.e., selecting valid and invalid translation pairs as in the baseline implementation, but not to be reversed.

[4] https://gitlab.com/kd-public/tiad-2017_baseline

– Two nodes are connected by a directed edge if they are given as source and target in the original data.
– For all languages except source and target language of the extrapolated dictionary, we also add an edge in the opposite direction.[5]
– Multiple nodes with the same attributes are unified. Thus, words which appear in several dictionaries are connected to several target words.

The baseline builds on cycles retrieved from this data, which may involve one or more pivot languages, e.g. 'skrivebord'@da → 'bureau'@fr → 'escrivaninha'@pt-BR → 'escritorio'@es → 'skrivebord'@da. The TIAD baseline implements a depth-first search for cycles over this data and returns the first cycle encountered. In our approach, we extract *all* possible cycles using a modified version of this approach. For the source and target languages of the dictionary to be inferred, we extract all paths from any source language word to any target language word. In the absence of manually devised gold data, we use these cycles and the paths as training and test data for the machine learning.

3 Approach

We use a simple Support Vector Machine (SVM) for classifying a source-word-target-word-pair as valid or invalid translation. In addition, we let the SVM determine both the likelihood of a pair belonging to the positive class and that of belonging to the negative class.

3.1 Defining training data

We define *positive instances* as being those pairs which occur in a cycle in which there is a direct connection (length 1) between the target word and the source word, i.e.,

1. we only consider pairs which occur in the provided dictionary as positive examples to be sure that the pair really is a valid translation[6], and
2. we do *not* consider every pair occurring in a cycle automatically as a valid translation.[7]

This approach formalizes the observation that the cycle criterion implemented in the baseline is likely to yield invalid translations in the case of correlated polysemy, cf. the following cycle where the correlated polysemy of Esperanto *pojno* and Spanish *muñeca* leads to a wrong translation pair 'doll'@en and 'Handgelenk'@de (i.e., 'wrist') [16]:

[5] This corresponds to the direction reversal as implemented by the baseline algorithm.
[6] For nl-fr, we therefore do not have positive examples for the SVN training, as there is no nl-fr dictionary in the test data.
[7] The baseline treats a pair as valid translation if at least one cycle is found during the depth-first search.

'doll' → 'poupée'@fr → 'pojno'@eo → 'Handgelenk'@de → 'muñeca'@es → 'doll'

We define *negative instances* as the word pairs which occur in a path and not in a cycle; these are in total 17 373 pairs. From these, we randomly sample 1 080 training instances, the same amount as the number of positive training examples.

3.2 Training features

For any given translation pair (w_{src}, w_{tgt}), we consider the following features:

Number of paths from w_{src} to w_{tgt} (NumP)
The existence of a high amount of such paths might indicate that the translation pair is a valid one as there are many possibilities how to get from the source word to the target word, varying in the amount, the succession and the language of the pivot words.

Frequency of source word in a dictionary (MaxOccDict)
Different translation possibilities of a source word in a dictionary means high polysemy for this word, making its translation more difficult. For words occurring in several dictionaries as source word, we take the maximum number of occurrence

Minimum/maximum path length (PLen)
A short path makes it less probable to change the meaning going from source to target word, a long path makes this in turn more likely; if there exists a really short path, that might be a good sign; we take both the minimum and maximum length of paths, as the maximum length might help the SVM to identify bad pairs

Difference of paths (PDiff)
We look at how the paths between w_{src} and w_{tgt} (P in the following) differ:
a) For every path $p \in P$, we retrieve the set of languages involved. We then count the number of sets (*language difference*).
b) For every path $p \in P$, we retrieve the set of words (nodes from a graph) involved. For every word set w, $P_w \subseteq P$ is the set of paths with the same word set. The number of switches $sw(P_w)$ is the number of paths in P_w, except that for paths for which also exists a reverse version, only one is being counted. If the same set of words occurs on different paths, this is likely to indicate reliable translation pairs. We return $\max_{P_w} sw(P_w)$ as the *word sequence difference excluding reversal*.

Minimum/maximum path probability (PProb)
For two words w_A and w_B with a direct connection in a dictionary $A \mapsto B$, we calculate their probability $P(w_A \to w_B) = |\{w'_B | w_A \to w'_B \in A \mapsto B\}|^{-1}$. The probability of any given path $p = w_0 \to w_1 \to ... \to w_{n-1} \to w_n$ between source word w_0 and target word w_n is $P(w_0 \to w_n) = \prod_{i=1}^{n} P(w_{i-1} \to w_i)$. We return minimum and maximum path probabilities.

Levenshtein distance on a path (Lev)

With the exception of Japanese, the provided dictionaries involve only two language families, Romance (French, Portuguese, Spanish) and Germanic (English, German, Dutch, Danish). As it is likely that cognates occupy the same semantic fields in languages descending from a common source, we calculate pairwise relative Levenshtein distance[8] for all words per language family on a path and return their average value as Levenshtein distance for each path from w_{src} to w_{tgt}. As a feature for the pair w_{src}, w_{tgt}, we take the average of all these path Levenshtein distances.

We employ this feature set for training an SVM classifier[9], and we also train SVMs with every feature in isolation to assess the impact of individual features on our gold data (valid translation pairs are pairs occuring in cycles and having a direct connection from target word back to source word, and invalid pairs are those which do not occur in a cycle). The classification is done by assigning the labels 1 or 0 to candidate translation pairs[10].

Table 1 lists precision, recall and F1 measure of our internal evaluation of SVM performance and individual features, using 80% of the data we defined as our gold data as training and the remaining 20% as test set. As mentioned above, our definition of gold data does not include nl→fr pairs, thus the results for this pair is not present in the table. The last row contains the scores if all features are used to train the SVM.

In general, classification performance for German and Portuguese are relatively low, with classification using the path length (PLen) being roughly on a par with the full feature set. Path length is a dominating factor for Danish and Spanish. For this translation pair, the combination of all features returns higher results than the features individually. For German and Portuguese, we get a higher (or equal (PLen)) precision when using all features for the SVM training compared to the use of an individual feature. However, the combination could not outperform the single features in terms of F1 and recall.

Reliable conclusions about the performance of individual factors apparently require a more substantial data, to counterbalance specific characteristics of individual dictionaries, to generalize beyond the apparent noise in the data and to avoid overfitting due to insufficient amounts of training data.

As a general pattern, it seems the Levenshtein-based approach performs poorly on both datasets. One reason may be found in the structure of the predefined paths where languages within one language family are usually adjacent. On longer or more homogeneous paths, Levenshtein may have a greater impact. Linguistic homogeneity (i.e., whether adjacent languages belong to the same language family) may actually explain why Levenshtein is more successfull for

[8] Relative Levenshtein is defined here as Levenshtein distance divided by the sum of the length of both strings.

[9] We use the C-SVM [3] implementation of libsvm [2], accessed via scikit-learn package for Machine Learning in Python [9] with RBF Kernel, $\gamma = 0.10$, $C = 1$, equally weighted classes and $\epsilon = 0.001$.

[10] Our submission results assigns probabilities, see below.

Danish and Spanish, as the shortest path between them involves only a single pivot language (French) which is historically related to Spanish, whereas the shortest German-Portuguese path connects both languages via English which is (due to Romance) influence rather remote from other Germanic languages. In this case, one may also ask whether limiting Levenshtein to pre-defined language families should not be extended to known language contact phenomena in order to accomodate the special ties between English and Romance.

A second aspect is that Levenshtein is actually a poor approximation for phonological similarity (which would be criterion for cognates and thus, semantic overlap) as all kind of character replacements are regarded equally likely, whereas sound change usually tends to preserve phonological characteristics (e.g., it is less likely that /o/ corresponds to /t/ in a related language than that it corresponds to /u/, but both substitutions are weighted equally in classical Levenshtein.) An alternative implementation with weighted Levenshtein would thus be advisable, however, most of the modifications, e.g. the Damerau-Levenshtein algorithm [4], use additional data to obtain the optimal weights, which may contradict to the nature of the closed track of the Shared Task, i.e. generating translation pairs without additional resources.

Finally, alternative strategies to aggregate Levensthein distance metrics may produce different results, too. It is possible that such different adaptations of Levenshtein would show a similar band-width in performance as the path-based metrics, so that our results reveal little about the applicability of form-based factors in general. Within the scope of the shared task, however, these could not be explored to a greater extend.

Table 1. Quality of the single features and their combination

	de ↦ pt-BR			**da ↦ es**		
	Precision	Recall	F1	Precision	Recall	F1
NumP	0.25	0.18	0.21	0.50	0.39	0.44
MaxOccDict	0.31	0.16	0.21	0.50	0.34	0.40
PLen	1.00	0.10	0.19	0.68	0.48	0.56
PDiff_a	0.21	0.21	0.21	0.48	0.48	0.48
PDiff_b	0.22	0.22	**0.22**	0.48	0.48	0.48
PProb	0.17	0.10	0.13	0.52	0.42	0.47
Lev	0.10	0.00	0.00	0.58	0.30	0.40
ALL	1.0	0.11	0.20	0.71	0.50	**0.59**

4 Results

After training the SVM model, we let the model predict the probability of a candidate translation pair being a valid translation. Following the baseline implementation, we thereby only consider pairs occurring in a cycle. We remove all the pairs for which the SVM returns a probability of less than 75%.

The TIAD Shared Task comes with two modes of evaluation — an evaluation against existing dictionaries as gold data (*Gold* in the table), and a manual assessment of precision on sample data instances (*Manual* in the table), both provided by KDictionaries. According to the the Shared Task results[11], our system outperforms the other participating systems in terms of manual and gold precision.

Table 2 lists the results, i.e. the precision values, of our system and the baseline implementation, as calculated by the organizers. Recall was not calculated, because it would require to determine all possible valid translations, which is a rather difficult endeavour and was not in the scope of the Shared Task[12]. For the 'Gold' evaluation, only inferences that were in the gold standard data were considered. The 'Manual' evaluation results were obtained by taking both the gold standard data and human translators' evaluation into consideration. For the 'Manual' evaluation, our system outperforms the baseline both for dk↦es and nl↦fr cases. It should be noted, however, that according to the evaluation on the gold standard data it is outperformed by the baseline for dk↦es and de↦pt-BR pairs. The reason behind this difference should be in some borderline cases which were not present in the existing dictionaries but were labeled as correct by a human annotator.

Table 2. Precision metrics for our system and the baseline implementation

	System		**Baseline**	
	Gold	Manual	Gold	Manual
dk ↦ es	0.40	**0.97**	**0.59**	0.89
nl ↦ fr	**0.57**	**0.74**	0.52	0.70
de ↦ pt-BR	0.30	0.94	**0.62**	**1.00**

An obvious reason for the good performance of the baseline (and, for that matter, our system) is that its prediction heavily relies on the existence of cycles in the data, by which the potential noise from polysemy is being elimitated relatively efficiently. This is a result very much in line with earlier research [13], however, it is also a slightly artificial scenario, as it is effectively only applicable for languages for which at least two bilingual dictionaries with two other languages already exist [5, 7]: In the practical reality of language documentation or NLP research on low-resource languages, for which bootstrapping inferred dictionaries would be particularly useful, normally only one major dictionary (between the minority language and the national language or English) is available, and if there are more, they are often limited in coverage (which limits the

[11] Cf. https://tiad2017.wordpress.com/data/.
[12] Justification according to the organizers' note regarding the evaluation results.

value of such languages as pivot languages).[13] Most related research therefore focuses on translation inference from simple paths rather than cycles [14, 15, 1, 10, 8]. Our internal evaluation (Tab. 1) which abstracts from this meta-factor indicates that path-based factors are successfully able to disentangle probable and less probable candidate translations as measured against the cycle criterion, but also that the combination of multiple path-based factors by means of machine learning is likely to outperform 'intuitive' metrics such as path probability.

Along with path-based factors, etymological closeness has been considered a major factor in such studies [12, 11]. Here, this factor has been approximated by a relative Levenshtein distance metric. The non-satisfying performance of this metric in our scenario has been discussed before, it should be noted, however, that conventional Levenshtein metrics are no longer considered to be state of the art, and more elaborate approaches to detecting cognates are to be tested in follow-up experiments.

A third category of features, involving semantic or grammatical information, requires external resources and was thus beyond our Shared Task contribution.

5 Conclusion

This paper described our contribution to the closed track of the Shared Task on *Translation Inference across Dictionaries* (TIAD-2017).

Further improvement of our approach is expected to be achieved by modification and inclusion of additional features for the SVM training. The features used here were mainly properties of the path from source to target word in the translation graph. A possible extension lies in the inclusion of word context features [17]: As the example phrases in the provided dictionaries only constitute a limited context, text corpora for each language should be consulted. Another possibility of including context information would be the usage of a distributional semantics approach. Word embeddings trained on different corpora would have to be mapped to a common semantic space in order to calculate vector distances.

In the overall evaluation, our system yields marginal (if any) improvement over the cycle-based based baseline. It is, however, more demanding in terms of resources: The exhaustive search for cycles and paths in the graph is computational expensive and for larger datasets therefore not feasible. However, as we used a machine learning approach, availability of massive training data is crucial and a computational acceptable alternative for the exhaustive search should be preferred.

It should be noted that the setup of the task and the provided data favors systems that make use of cycles or loops in translation chains. For low resource languages, where translation inference across dictionaries is probably most relevant, this scenario is, however, rather unlikely, as multiple, large-coverage dic-

[13] As a representative example for such low-resource dictionaries, one may consider the Intercontinental Dictionary Series (IDS) which provides data for up to only 1310 (!) entries per language [6].

tionaries are mostly available for major languages. Our observations regarding the impact of path-based factors (and possible limitations of Levensthein-based methods) are nevertheless also relevant for the more general case where only non-cyclical sequences of dictionaries are available. For future editions of the Shared Task, we would thus be interested in exploring a path-based rather than cycle-based setup for translation inference across dictionaries.

Acknowledgments

The research described in this paper was conducted in the project 'Linked Open Dictionaries' (LiODi, 2015-2020), funded by the German Ministry for Education and Research (BMBF) as an Early Career Research Group on eHumanities.

References

1. Bond, F., Ogura, K.: Combining linguistic resources to create a machine-tractable Japanese-Malay dictionary. Language Resources and Evaluation 42(2), 127–136 (2008)
2. Chang, C.C., Lin, C.J.: LIBSVM: A library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 1–27 (2011)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
4. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM 7(3), 171–176 (Mar 1964), http://doi.acm.org/10.1145/363958.363994
5. István, V., Shoichi, Y.: Bilingual dictionary generation for low-resourced language pairs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2009). pp. 862–870. Edinburgh, Scotland (2009)
6. Key, M.R., Comrie, B. (eds.): Intercontinental Dictionary Series (IDS). Max Planck Institute for Evolutionary Anthropology, Leipzig (2015), http://ids.clld.org/
7. Lam, K.N., Al Tarouti, F., Kalita, J.K.: Automatically creating a large number of new bilingual dictionaries. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-2015). pp. 2174–2180. Austin, Texas (2015)
8. Mairidan, W., Ishida, T., Lin, D., Hirayama, K.: Bilingual dictionary induction as an optimization problem. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014). pp. 2122–2129. Reykjavik, Iceland (2014)
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12(Oct), 2825–2830 (2011)
10. Saralegi, X., Manterola, I., Vicente, I.S.: Analyzing methods for improving precision of pivot based bilingual dictionaries. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2011). p. 846–856. Edinburgh, Scotland (2011)
11. Schulz, S., Markó, K., Sbrissia, E., Nohama, P., Hahn, U.: Cognate mapping: A heuristic strategy for the semi-supervised acquisition of a Spanish lexicon from a Portuguese seed lexicon. In: Proceedings of the 20th International Conference

on Computational Linguistics (COLING-2004). pp. 813–819. Geneva, Switzerland (2004)
12. Skoumalova, H.: Bridge dictionaries as bridges between languages. International Journal of Corpus Linguistics 6(11), 95–105 (2001)
13. Soderland, S., Etzioni, O., Weld, D.S., Skinner, M., Bilmes, J., et al.: Compiling a massive, multilingual dictionary via probabilistic inference. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the Association of Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009). p. 262–270. Suntec, Singapore (2009)
14. Tanaka, K., Umemura, K.: Construction of a bilingual dictionary intermediated by a third language. In: Proceedings of the 15th Conference on Computational Linguistics (COLING-1994). p. 297–303. Stroudsburg, PA, USA (1994)
15. Tsuchiya, M., Purwarianti, A., Wakita, T., Nakagawa, S.: Expanding Indonesian-Japanese small translation dictionary using a pivot language. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. pp. 197–200. Prage, Czech Republic (2007)
16. Villegas, M., Melero, M., Bel, N., Gracia, J.: Leveraging RDF graphs for crossing multiple bilingual dictionaries. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC-2016). Portorož, Slovenia (2016)
17. Yujie, Z., Ma, Q., Isahara, H.: Automatic construction of Japanese-Chinese translation dictionary using English as intermediary. Journal of Natural Language Processing 12(2), 63–85 (2005)