WORDNETS SAVE THE DAY (YET AGAIN!) : COGNATE DETECTION FOR INDIAN LANGUAGES

Diptesh Kanojia | Kevin Patel | Pushpak Bhattacharyya | Malhar Kulkarni | Reza Haffari
IITB – Monash Research Academy
Center For Indian Language Technology (CFILT)

Please send any and all questions to:

diptesh@cse.iitb.ac.in

COGNATES ARE YOUR FRIENDS

Words which have a common etymological origin due to a diachronic relationship across multiple languages (Crystal, 2008).

Such word pairs share a semantic affinity and can facilitate the foreign language learning process.

In short, it would be easier to learn a non-native language.

In terms of Natural Language Processing (NLP), computational tasks like Machine Translation, Information Retrieval, and Computational Phylogenetics **can benefit from Automated Cognate Detection** (Al-Onaizan et al., 1999; Meng et al., 2001; Rama et al., 2018).

BUT WATCH WHO YOUR CALL A 'FRIEND'!

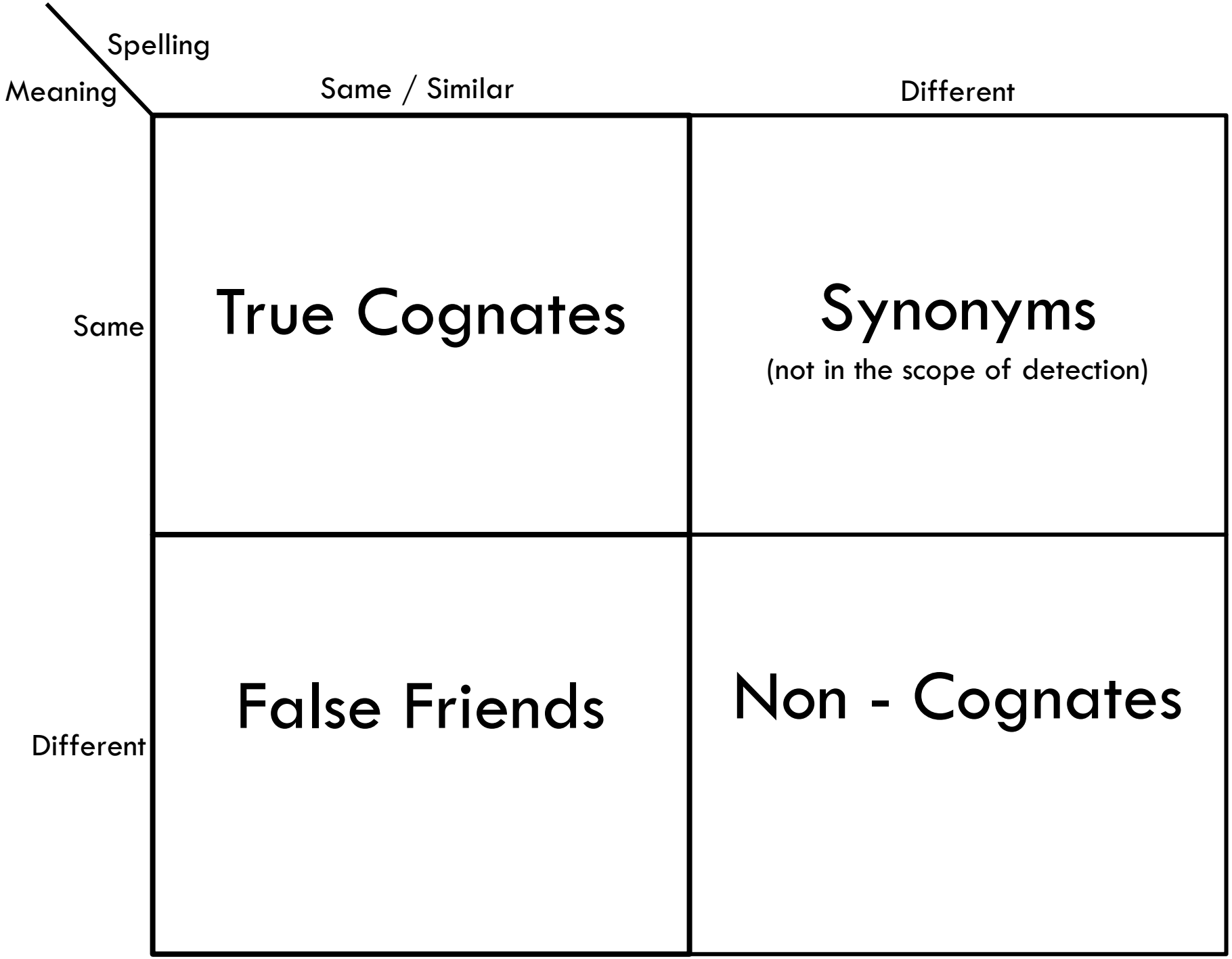
..and then there are 'False Friends' which, on the surface look the same, but mean different!

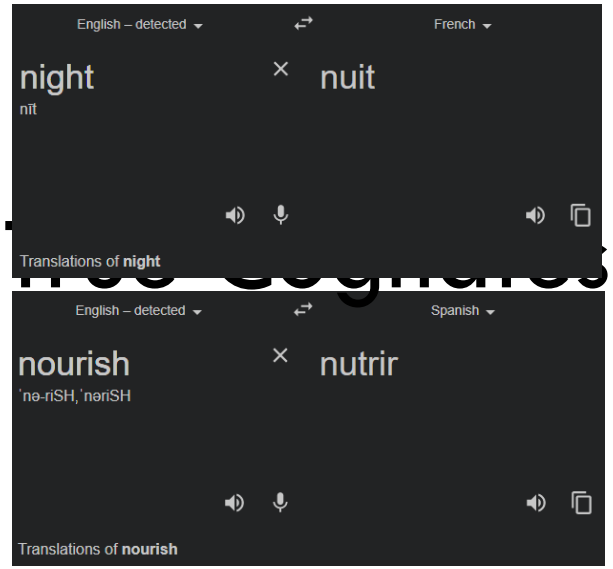
False friends segregation from among these cognates should be an important part of any cognate detection methodology as orthographic (spelling-based) or phonetic similarity (sound correspondences) based methods can also overgeneralize just like any human second language learner.

Indian language pairs borrow a large number of cognates (thus, also false friends) due to their shared ancestry of these languages.

Cognate detection has already been applied in NLP for Sentence alignment (Simard et al.,1993; Melamed,1999) and inducing translation lexicons (Mann and Yarowsky, 2001; Tufis, 2002).

Meaning \ Origin	Same	Different
Same	**True Cognates** **Father – Père** (En – Fr) **हज़ार – हाजार** (Hi – Bn) (hazaar – hajaar) (both meaning "thousand") **जीवन – जीबन** (Hi – Bn) (Jeevan – jeeban) (both meaning "life") **Celebrate – Celebrar** (En – Es) (both meaning the "action of celebrating")	**False Cognates** **ache – *ákhos*** (En – El) (both meaning "pain") **Saint – Sant** (En – Sa) (both meaning "a holy person") ***feu - Feuer*** (Fr – De) (both meaning "fire") ***ciao – chào*** (It – Vi) (both meaning "hello/goodbye")
Different	**False Friends** ***friend - frände*** (En – Sv) (meaning "friend" and "Relative" respectively) ***Friend – frænde*** (En - Da) (meaning "friend" and "Relative" respectively) **Vase – Vaso** (En - Es) ("flowers holder" and "glass of water") **अभिमान – ओभिमान** (Hi - Bn) (obhimaan – abhimaan) (both meaning the "action of celebrating")	**Non Cognates** **sentences - palabras** (En – Es) **enemy – bạn** (En – Vi) **comma – kochać** (En - Pl) **Bank – bank** (En - En) (When both mean differently – context wise)


Spelling
Meaning
Same / Similar
Different
Same
Different
True Cognates
Synonyms
(not in the scope of detection)
False Friends
Non - Cognates

Meaning \ Spelling	Same / Similar	Different
Same		Synonyms (not in the scope of detection)
Different	False Friends	Non - Cognates

SCRIPT STANDARDIZATION

The languages used in this task not all belong to the same script.

To find the surface similarity using the conventional approaches like Normalized Edit Distance (NED) based Similarity (1 – NED), Cosine Similarity (CoS), and Jaro-Winkler Similarity (JWS); we convert all the other scripts to Devanagari script which is used in Hi, Mr, Ne and Sa languages.

We use the Indic-NLP Library to perform Unicode offsetting to convert the other scripts to Devanagari script.

We always use Hindi as the source side languages since it provides ease in validating our output.

DATASETS

Wordnet Dataset (WNData)

IndoWordnet (Bhattacharyya, 2017)
(18 languages)

We use 11 languages and extract word pairs from *un*-linked synsets.
[Hi, Mr, Pa, Gu, Bn, Sa, Ne, Ml, Ta, Te, Ur]

We extract word pairs which are similar in spelling based on a weighted lexical similarity (WLS) score.

Parallel Corpus Dataset (CData)

ILCI Parallel Corpus (Jha, 2010)

We use the same 11 languages and extract all word pairs from parallel lines.

We use the same methodology to prepare this word-pair dataset.

(In case of unavailable parallel corpus for a language pair, we crawled the web for parallel corpus)

SIMILARITY MEASURES

We come up with a weighted measure by combining the conventional similarity scores.

The conventional similarity measures we use are NED based Similarity score, Cosine Similarity and Jaro-Winkler Similarity.

We ran experiments for providing weights our lexical similarity equation and empirically decided to provide 50% weight to NED, 25% to CoS and 25% to JWS.

The final weighted measure looks like:

$$\mathbf{WLS = 0.5*NED + 0.25*CoS + 0.25*JWS}$$

OUR APPROACH (WEIGHTED LEXICAL SIMILARITY BASED)

We use the weighted Orthographic / lexical similarity (WLS) score as ***our approach to find our lexical similarity between word pairs from the same synset.***

To compute the score for word pairs between the nine language pairs, we compare every word in the parallel synsets. For parallel corpus, we compute the score for every word pair in the parallel lines.

If the **WLS score for a word pair is > 0.5, we provide it a positive label** and add it to the training data; for scores **< 0.5 we provide the word pair a negative label** and add it to the training data.

We decide on this threshold empirically since we also tried using 0.25, 0.60 and 0.75 but based on the training performance, we select 0.5 as the threshold.

CLASSIFIERS – FEED FORWARD NETWORK

We use two different approaches to train a classification model. Here, we describe the simple feed forward neural network classification model (FFN)

1. We use a simple Feed Forward Neural network and treat the word as a whole.

2. The source side and target side words reside in separate embedding spaces.

3. The target word passes through the target embedding layer and the output of both embedding lookups is concatenated.

4. The resulting representation is passed to a fully-connected layer with ReLU activation followed by a *softmax* layer.

CLASSIFIERS – RECURRENT NEURAL NETWORK

Here, we describe the recurrent neural network classification model (RNN)

1. Here, we treat the word as a sequence of characters.

2. The embedding spaces contain characters from the source and the target side.

In a similar fashion, the source and target side characters pass through their respective embedding layers and at the end the output is concatenated.

The resulting representation is passed to a fully-connected layer with ReLU activation followed by a *softmax* layer.

RESULTS

We can clearly see that the classifiers trained on the WNData are performing better for both FFN and RNN.

We also observe that RNN outperforms FFN uniformly and with significant margins as is uses character-based embeddings to train.

The highest 5-fold evaluation score achieved was for the classification models on the language pair Hindi-Sanskrit (i.e., 91.66) which are very closely related and contain a lot more similar words.

	FFN		RNN	
	D1	D2	D1	D2
Hi-Mr	69.76	85.76	74.76	89.78
Hi-Bn	65.18	81.04	69.18	86.44
Hi-Pa	73.04	78.50	76.04	83.64
Hi-Gu	61.74	79.16	69.84	89.44
Hi-Sa	61.72	85.87	68.92	91.66
Hi-Ml	56.96	74.77	66.96	79.59
Hi-Ta	55.62	61.70	65.62	68.92
Hi-Te	52.78	65.26	62.78	74.83
Hi-Ne	70.20	83.85	80.20	89.63
Hi-Ur	69.99	73.84	76.99	80.12

Table 1: Stratified 5-fold evaluation using Deep neural models on PCData (D1) and WNData (D2)

CONTRIBUTION OF THE WORDNETS

Without the use of the Wordnet model where words lie in the same sematic space in a synset, we would not have been able to achieve these results.

We also wanted to check if WNData can somehow improve the scores for the PCData dataset.

We added 20% chunks of WNData to PCData to check if this can somehow improve the results and re-trained the classifiers for each language pair.

As a result, after adding 80 to 100% of WNData to PCData, we were able to achieve significant improvements in the overall results.

RESULTS ON OVERALL DATA

	Corp+WN20		Corp+WN40		Corp+WN60		Corp+WN80		Corp+WN100	
	FFN	RNN	FFN	RNN	FFN	RNN	FFN	RNN	FFN	RNN
Hi-Mr	70.12	74.12	73.56	78.37	76.09	81.56	81.34	85.24	86.90	91.87
Hi-Bn	71.06	73.17	73.29	74.98	77.33	76.28	83.99	81.45	82.18	89.58
Hi-Pa	74.16	75.94	76.02	77.39	76.18	79.04	78.04	81.22	80.66	85.64
Hi-Gu	65.26	70.76	71.21	74.83	75.09	79.95	80.14	84.32	81.85	89.81
Hi-Sa	65.93	74.23	69.25	77.51	74.84	79.92	81.03	86.62	88.13	93.86
Hi-Ml	57.75	59.38	56.31	65.67	58.02	71.19	61.01	75.59	69.11	82.54
Hi-Ta	54.63	60.12	56.69	63.38	57.46	66.17	59.36	67.17	60.41	70.62
Hi-Te	53.21	58.18	56.19	63.90	64.15	67.70	65.19	70.65	66.10	74.92
Hi-Ne	70.78	71.23	74.30	78.11	72.19	83.20	79.70	85.01	84.69	90.95
Hi-Ur	69.94	71.25	70.01	72.35	72.03	76.59	71.07	78.27	73.99	80.99

Table 2: Results after we append WNData to PCData in chunks of 20%.

CONCLUSION

We investigate the cognate detection task for Indian language pairs (Hi-Bn, Hi-Gu, Hi-Pa, Hi-Mr, Hi-Sa, Hi-Ml, Hi-Ta, Hi-Te, Hi-Ne, and Hi-Ur).

We use script converted Wordnet data (WNData) and Parallel Corpus Data (PCData) for the task of cognate detection.

Using our approach which focuses on taking the help of Wordnets to ensure two words are related semantically, we apply a weighted lexical similarity measure to compute word pairs which can be potentially True Cognates.

We use two classifiers (word-based FFN and character-based RNN) and build models to classify word pairs as true cognates and show that RNNs significantly beat FFNs when performing the task. We also show that WNData shows much better results compared to PCData.

We perform another experiment to check for the contribution of WNData and add chunks of 20% WNData to PCData and show significant improvements thus showing that WNData can actually help the task of cognate detection.

FUTURE WORK

In future, we aim to investigate the task of cognate detection using CNNs and also build a standard gold dataset which can help us provide F-scores and thus validate our methodology better.

We also aim to use cross-lingual word embeddings for the task of cognate and false friends detection from the Wordnet data.

We would also like to include the other Indian languages in the dataset (currently, the availability of parallel corpus for those langauges was the issue).

We would also like to see if the detected cognates can somehow help NLP applications such as Machine Translation for Indian languages and Computational Phylogenetics.

THANK YOU

Again,

Please send any and all questions to:

diptesh@cse.iitb.ac.in

REFERENCES

DA Crystal. 2008. Dictionary of linguistics and phonetics 6th edition crystal. DA Crystal – Oxford: Blackwell Publishing.

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A Smith, and David Yarowsky. 1999. Statistical machine translation. InFinal Report, JHU Summer Workshop, volume 30.

Helen M Meng, Wai-Kit Lo, Berlin Chen, and Karen Tang. 2001. Generating phonetic cog-nates to handle named entities in english-Chinese cross-language spoken document retrieval. In IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01., pages311–314. IEEE.

Taraka Rama, Johann-Mattis List, Johannes Wahle, and Gerhard Jäger. 2018. Are automatic methods for cognate detection good enough for phylogenetic reconstruction in historical linguistics? arXiv preprint arXiv:1804.05416.

Michel Simard, George F Foster, and Pierre Isabelle. 1993. Using cognates to align sentences in bilingual corpora. In Proceedings of the 1993conference of the Centre for Advanced Studies on Collaborative research: distributed computing-Volume 2, pages 1071–1082. IBM Press.

MORE REFERENCES

I Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. Computational Linguistics, 25(1):107–130.

Gideon S Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, pages 1–8. Association for Compu-tational Linguistics.

Dan Tufis. 2002. A cheap and fast way to build useful translation lexicons. In COLING 2002:The 19th International Conference on Computational Linguistics.

Pushpak Bhattacharyya. 2017. Indowordnet. In The WordNet in Indian Languages, pages 1–18.Springer.

Girish Nath Jha. 2010. The tdil program andthe indian langauge corpora intitiative (ilci). In LREC.

Piotr Bojanowski, Edouard Grave, Armand Joulin,and Tomas Mikolov. 2017. Enriching word vec-tors with subword information. Transactions of the Association for Computational Linguistics,5:135–146.