Proceedings of 1st Workshop on Language Technologies for Historical and Ancient Languages, pages 79–83
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Comparing Statistical and Neural Models for Learning Sound Correspondences

Clémentine Fourrier, Benoît Sagot
Inria - ALMAnaCH
2 rue Simone Iff, 75012 Paris, France
{clementine.fourrier, benoit.sagot}@inria.fr

Abstract

Cognate prediction and proto-form reconstruction are key tasks in computational historical linguistics that rely on the study of sound change regularity. Solving these tasks appears to be very similar to machine translation, though methods from that field have barely been applied to historical linguistics. Therefore, in this paper, we investigate the learnability of sound correspondences between a proto-language and daughter languages for two machine-translation-inspired models, one statistical, the other neural. We first carry out our experiments on plausible artificial languages, without noise, in order to study the role of each parameter on the algorithms respective performance under almost perfect conditions. We then study real languages, namely Latin, Italian and Spanish, to see if those performances generalise well. We show that both model types manage to learn sound changes despite data scarcity, although the best performing model type depends on several parameters such as the size of the training data, the ambiguity, and the prediction direction.

Keywords: Cognate prediction, Proto-form prediction, Statistical models, Neural models

1. Introduction

Since the works of the Neogrammarians (Osthoff and Brugmann, 1878), it is assumed that the lexicon of a language evolves diachronically according to regular sound changes, notwithstanding morphological phenomena, lexical creation and borrowing mechanisms.

The regularity of sound change can be modelled as follows. If, at a given "point" in time, a phone (or phoneme) in a given word changes into another phone (or phoneme), then all occurrences of the same phon(em)e in the same context change in the same way.[1] Such a global change is modelled as a *sound law*. The phonetic history of a language from an earlier to a later stage can then be modelled as an ordered sequence of sound laws. Sound laws are usually identified by studying *cognates*: given two languages with a common ancestor, two words are said to be cognates if they are an evolution of the same word from said ancestor, called their *proto-form*.[2,3] Therefore, the phonological differences between two cognates, which can be modelled as a sequence of *sound correspondences*, capture some of the differences between the phonetic evolution of the languages.

Most methods for sound correspondences identification start by aligning sequences of characters or phones, to which they then apply statistical models, clustering methods, or both (Mann and Yarowsky, 2001; Inkpen et al., 2005; List et al., 2017; List et al., 2018; List, 2019) with the notable exception of Mulloni (2007), who uses Support Vector Machines. However, this task presents a number of similarities with machine translation (MT), as they both involve modelling sequence-to-sequence cross-lingual correspondences,[4] yet state-of-the-art neural network techniques used in MT (Bahdanau et al., 2015; Sutskever et al., 2014; Luong et al., 2015) have only been used once for sound correspondence prediction, with disappointing results (Dekker, 2018).

Our goal in this paper is to study under which conditions either a neural network or a statistical model performs best to learn sound changes between languages, given the usually limited available training data.[5] We first compare the performances of these two types of models in an ideal setting. To do that, we generate an artificial phonetised trilingual lexicon between a proto-language and two daughter languages, use it to train each model with varying hyperparameters and compare the results. We observe that statistical models perform better on small data sizes and neural models on cases of ambiguity. We then present the results of preliminary experiments, reproducing the same study under real life conditions, using a trilingual cognate dataset from Romance languages. We observe that both models learn different kind of information, but that it is too early to conclude; experiments need to be extended with better and bigger datasets.

2. Data

2.1. Artificial Data Creation

In order to compare how both model types perform on the task of sound correspondence learning in an ideal setup, we create an artificial lexicon, composed of a proto-language and its reflect in two artificially defined daughter languages. Using artificial data for such a proof of concept offers several advantages: we can investigate the minimum number

[1] For example, the sequence [ka] in Vulgar Latin changed into [ʧa] in Old French, then to [ʃa] in French. This is illustrated by *chat* [ʃa] 'cat' < Vulg. Lat. *cattus* *[kat.tʊs] and *blanche* [blɑ̃ʃ] 'white (fem.)' < *blanca* *[blan.ka].

[2] For example, Pol. *być* 'to be', Cz. *být* 'id.' and Lith. *būti* 'id.' are cognates as they share the same Proto-Balto-Slavic ancestor.

[3] The term 'cognate' is sometimes used with broader definitions that are tolerant to morphological differences between the proto-forms of both words and/or to morphological restructurings in the history of the languages.

[4] MT generally process sequences of (sub)words, whereas we process sequences of phon(em)es.

[5] Such a method could also be applied to learn orthographic correspondences between close languages, provided said correspondences are regular enough; however, this is not the point of this paper as we focus on an historical linguistic application.

of word pairs required to successfully learn sound correspondences, as well as control the different parameters constraining the proto-language (number of phonemes, phonotactics) and its transformation into the daughter languages (e.g. number of sound changes). However, the artificial data must be realistic, to not impair the linguistic validity of the experiment; the proto-language must have its own realistic phonology, obey phonetic and phonotactic rules, and its daughter languages must have been generated by the sequential application of plausible sound changes.

Creating a Proto-Language We create an algorithm which, given a phone inventory and phonotactic constraints[6], generates a lexicon of a chosen size.[7]

For our experiments, we draw inspiration from Latin and Romance languages. More precisely, we use:

- The phone inventories of Romance languages: each lexicon generated uses all the phones common to all Romance languages, as well as a randomly chosen subset of less common Romance phones.[8]
- The phonotactics of Latin, as detailed in the work of Cser (2016): each word is constructed by choosing a syllable length in the distribution, and its syllables are then constructed by applying a random set of the corresponding positional phonotactic rules.

Generating a Daughter Language Given the proto-language, we create a daughter language by, first, randomly choosing a set of sound changes, then consecutively applying each chosen sound change to all words in the lexicon. Among the main possible sound changes for Romance languages are apocope, epenthesis, palatalisation, lenition, vowel prosthesis and diphtongisation. The dataset generated for this paper used two sets, each of 15 randomly chosen sound changes, to generate two daughter languages. Two examples from our generated dataset are [stra] > [isdre], [estre] and [ʒolpast] > [ʒolbes], [ʒolpes].

2.2. Real Dataset Extraction

Our second goal being to study how our results in an artificial setting generalise to a real-life setting, we need to gather a dataset of related real languages, from a well known direct ancestor language to two closely related but different daughter languages. We choose to study Latin (LA) as the ancestor language, with Italian (IT) and Spanish (ES) as its daughter languages.

Raw Data Extraction EtymDB 2.0 (Fourrier and Sagot, 2020) is a database of lexemes (i.e. triples of the form ⟨language, lemma, meaning expressed by English glosses⟩), which are related by typed etymological relations, including the type "inherited from." To generate the cognate dataset from EtymDB, we followed the inheritance etymological paths between words; two words form a cognate pair if they share a common ancestor[9] in one of their common parent languages (Old Latin, Proto-Italic, or Proto-Indo-European for LA-IT and LA-ES, Vulgar Latin, Latin, and the previous languages for IT-ES).

Phonetisation and filtering The phonetisation of the real data is done using Espeak, an open source multilingual speech synthesiser (Duddington, 2007 2015), which can also convert words or sequence of words into their IPA representations. We phonetise each word independently, then add to each phonetised word a start-of-sentence token indicating their language and a generic end-of-sequence token (EOS), following Sutskever et al. (2014).[10] When faced with competing pairs, i.e. pairs whose source word is the same but whose target words differ, we only retain the pair with the lowest Levenshtein edit distance (method with the strongest cognate recall according to List et al. (2018)).

2.3. Datasets properties

The artificial dataset contains 20,000 unique word triples containing a proto-language (PL) word and its reflects in the two daughter languages (DL1 and DL2). Samples of various sizes are then randomly drawn from this dataset.

The real-life dataset contains 605 cognate triples for LA-ES-IT (1/3-2/3 split in training and test set) as well as 388 additional cognate pairs for LA-IT, 296 for LA-ES, and 1764 for IT-ES, all extracted from EtymDB-2.0 (see above). Early experiments on real-life data have shown that, to compensate for noise, monolingual data must be used to constrain the encoder and decoder of each language to learn what can be plausible phone sequences in a word. We therefore extract 1000 words for each language.

3. Experimental Setup

Task Description For each dataset available, we want to compare how well the statistical and neural models learn sound correspondences between related languages. We define the corresponding task as the translation of phonetised cognates from one language to another.

However, we expect said translation tasks to vary considerably in terms of difficulty: since several proto-forms can give the same daughter form, going from a daughter language to a mother language should be harder than the opposite. To account for this ambiguity, we predict 1, 2, and 3-best answers with each model.

Statistical Model Moses is the reference (open source) tool for statistical MT (Koehn et al., 2007). We first tokenise and align the bilingual data using GIZA++ (Och and Ney, 2003), then train a 3-gram language model of the output (Heafield, 2011), a phrase table that stores weighted correspondences between source and target phonemes (we use 80% of our training data) and a reordering model. The relative weights of the language model, the phrase table and the reordering model are then tuned (we use MERT) on development data, the remaining 20% of our training data. For a given input, the decoder can then find the highest scoring equivalent(s) of an source word in the target language.

[6] Phonotactics govern which phonemes sequences are allowed.

[7] Code available at https://github.com/clefourrier/PLexGen

[8] For example, vowels common to all Romance languages are [a] [e] [i] [o] [u], and a subset of extra vowels could be [ɔ] [ɛ] [ɪ]

[9] Said ancestors are those present in the database, not an exhaustive list of all possible cases

[10] In the decoding phase of the model, everything predicted after an EOS token is discarded.

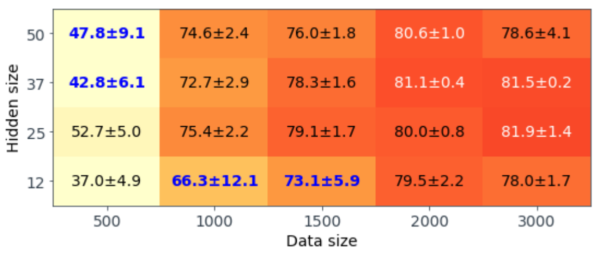


Figure 1: BLEU scores for MEDeA, function of the data size and hidden dimension, for the PL→DL1 pair.

Neural Model MEDeA (Multiway Encoder Decoder Architecture) is our implementation of one of the classical approaches in neural MT: the sequence-to-sequence encoder-decoder model with attention (Bahdanau et al., 2015; Luong et al., 2015).[11] We use an architecture with a specific single layer encoder and a specific single layer decoder for each source and each target language. We use an attention mechanism specific to each decoder (and not encoder-dependent). For a given multilingual lexicon the model learns on all possible language pairs,[12] which constrains the hidden representation to a single space. For all experiments, each phone is embedded as a vector of length 5,[13] and MEDeA is trained with batches of size 30, a batch dropout of 0.2, no layer or attention dropout, and Adam optimisation with a 0.01 learning rate.

Evaluation Metric We use BLEU as an evaluation metric.[14] BLEU is based on the proportion of 1- to 4-grams in the prediction that match the reference. This is extremely interesting for our task, as sound changes can affect several succeeding phones: this score gives us, not only the character error rate computed by the 1-gram, but also the errors in the phone successions computed by the 2- to 4-grams in BLEU. A major criticism of the BLEU score for MT is that it can under-score correct translations not included in its reference set. This does not apply in our case, since there is only one possible "translation" of a word into its cognate in another language.
In order to use BLEU even when we produce n>1 "translations", we compute BLEU scores by providing the n-best results as the reference, and our input word as the output.

4. Experiments on Artificial Data

4.1. Model Parameters

For all our experiments on artificial languages, we train the models on our multilingual artificial lexicon.

MEDeA learns a single model for all possible language pairs, on 50 epochs. We train it with hidden dimensions of 12, 25, 37, and 50, training set sizes of 500, 1000, 1500, 2000, and 3000 triplets of words, for 1, 2 or 3 best results. To limit the impact of train/test set separation, we repeat these experiments using three different shuffling seeds.

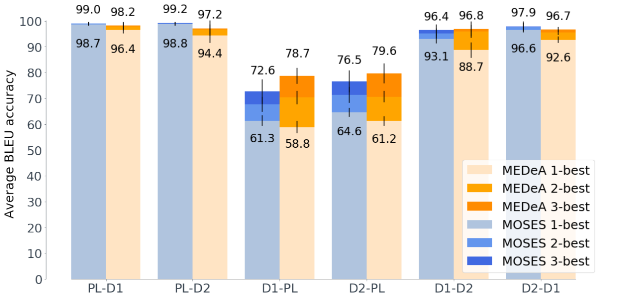


Figure 2: BLEU scores averaged over all runs for all training data sizes (except 500).[15]The bottom part of each bar represents the BLEU score of the most probable predicted word for each input word. The mid (resp. top) part of each bar corresponds to the gain in BLEU obtained by also considering the second-best (resp. third-best) ranked prediction for each input word.

MOSES is trained on the same data splits as MEDeA, shuffled in the same order, to predict 1 to 3 best results. However, we have to do one run for each language pair, as MOSES can only learn on bilingual data.

4.2. Impact of the Hidden Dimension on Neural Models

We study the impact of the hidden dimension on the performance of MEDeA. No matter the data size, we observe in Figure 1 that a hidden dimension of 12 is consistently too small to learn as well as the rest, and very sensitive to instability (see the std in blue). A hidden dimension of 50 only performs well with big enough data sets, and is very sensitive to instability below 1000 pairs of words. On average, the hidden dimension which achieves the best performance for the data sizes we have is 25, as it represents a good balance between a high enough complexity of representation and a small enough number of weights to learn with. For this reason, in the rest of the paper, we will only introduce the results corresponding to a hidden dimension of 25 for the neural network.

4.3. Model Independent Observations

This analysis focuses on data sizes of 1000 and above, as the impact of very small datasets (500 word pairs per language) on the prediction BLEU scores of both MOSES and MEDeA will be specifically discussed in the next section.
Across all experiments and models, we observe in Figure 2 that the easiest situation to learn is the predict from the proto language (PL) to its daughters (98 BLEU), then from one daughter language to the other (92-95 BLEU), and that, finally, the hardest task by far is to go from a daughter language to its mother (60-75 BLEU): there is a difference of 20 points between the best results from mother to daughter and the best from daughter to mother.

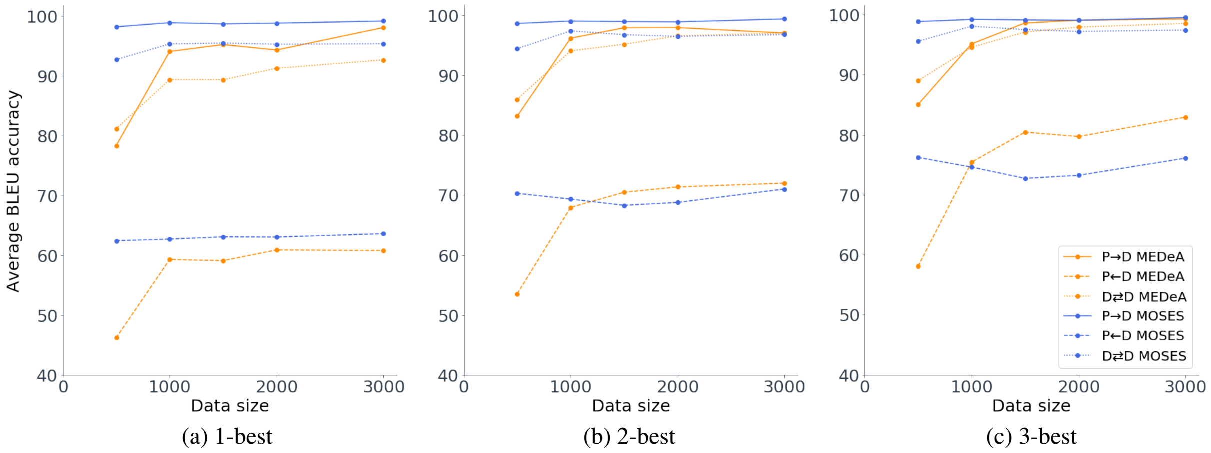
[11] Code available at https://github.com/clefourrier/MEDeA

[12] For example, for a bilingual Spanish-Italian lexicon, the model will learn on Spanish to itself, Italian to itself, Spanish to Italian and vice versa.

[13] The embedding size was chosen in preliminary experiments, and was the best choice between 2, 5 and 10. This seems adequate relative to the total vocabulary size, of less than 100 items

[14] We use SacreBLEU, Post (2018)'s implementation

[15] Results obtained with a data size 500 skew the average considerably, being several standard deviations apart from the others, for reasons discussed in Section 4.2., and were thus removed.



(a) 1-best

(b) 2-best

(c) 3-best

Figure 3: BLEU scores for the n-best prediction, for all experiments.

Along the same lines, we also observe that using 2 or 3 best experiments barely improves the result for the first two situations (adds 2 to 5 points from 1-best to 3-best on average), when it considerably increases the BLEU score for the prediction from daughter to mother language (20 to 25 points for MEDeA, 10 to 15 points for MOSES). This difference, due to ambiguity, was expected, and described in the experimental setup.

4.4. Comparison Between Models

Both models can learn to predict in all directions, but they perform well under different circumstances (Figure 3).

1-best Experiments On 1-best experiments, the statistical model consistently outperforms the neural model, though not by far when reaching data sizes of 2000 and above.

***n*-best Experiments** With very little data (500 word pairs), the statistical model is significantly better; the neural model overfits on too little data. However, with a lot of data (2000 word pairs per language and above), the neural model outperforms the statistical model. This difference in performance seems to come from the better modelling of language structure by the neural model, as will be discussed in Section 5.2..

With 1000 and 1500 training pairs, the performance is roughly equivalent between the two models for 2 and 3 best (the statistical model is slightly better on 1000 word pairs, the neural network slightly better on 1500 word pairs).

5. Preliminary Experiments on Real Data

5.1. Model Parameters

To assess whether our results transfer to real world data, we carried out preliminary experiments on our real datasets. We expect both models to perform worse than on artificial data, since real data can contain noise, both from extraction errors and linguistic phenomena.

MEDeA is trained with the real dataset, on all language combinations possible (IT, ES, LA) at once, with early stopping at 50 epochs. We train it for 3 shuffling seeds, comparing a hidden size of 12 to 50, and 1, 2 or 3 best results, this time using all the data we have.

MOSES is trained on pairs of language combinations separately. We provide it with the same data splits, with the exception of monolingual data, removed from its training set. The triplets of manually corrected data is treated as several sets of pairs, for the same reasons.

Impact of Data Size on Neural Network Optimal Hyperparameters As mentioned in the data descriptions, not all language pair datasets are the same size. There are about 600 word pairs for ES-LA, 700 for IT-LA, and 2.5 times that for ES-IT. We observe that for low resource pairs, the corresponding best hidden size is 25, when for almost 2000 pairs, the best hidden size is 50, confirming what was observed in artificial data experiments. We will systematically investigate in further work the impact of data size on the best hidden dimension for learning.

5.2. Results

General Comparison We observe that, on this set of real data, the statistical model systematically outperforms the neural network, by on average 15 points. Neural networks are highly sensitive to noise and data inconsistencies when trained with too little data, especially without layer dropout.

Impact of the Data Size For our smallest dataset, ES-LA, BLEU scores ranges from 18 to 33 for MEDeA, and from 29 to 47 for MOSES (1-best to 3-best); for our biggest dataset, ES-IT, BLEU scores ranges in both direction from 40 to 54 for MEDeA, and 50 to 64 for MOSES (1-best to 3-best). Even for MOSES, there is a size threshold under which learning is significantly difficult.

What Are the Models Learning? When looking at the respective 3-best predictions of the two models, we observe that the statistical model learns sound correspondence patterns when the neural network learns the underlying structure of the data. For example, for IT→LA, the neural network consistently predicts several forms as possible words translations: [rustiko] ‘rustic’, coming from [rʊstɪkʊs] ‘of the country’, is predicted as [rʊstɪkʊs] (masc.), [rʊstɪkʊm] (neut.), and [rʊstɪkss] (nonsense) by MEDeA, vs [rʊkʊstɪ], [rʊɪkɔst] and [ʊsrtɪkwʊs], three meaningless forms by

MOSES.[16] It even allowed us to identify errors in our data: [ramo] ‘branch’ < [ramʊs] ‘branch’, erroneously related to Latin [radɪks] ‘root’ (cognate with [ramʊs]) in our dataset, was predicted by MEDeA as [ramʊs] (masc.), [ramʊ], [ramʊm], and by MOSES as [mʊr], [rɛam], and [raɛm].

6. Conclusion

Through this paper, we studied the respective performances of a statistical and a neural model, in two different settings, to produce the directly related correspondent of a source language word in a related, target language (i.e. to predict the cognate of a source word in a sister language of the source language, the etymon of a source word in a parent language, or the reflex of a source word in a daughter language). Our experiments with artificial data allowed us to study both models in a controlled setting. We observed that statistical models perform considerably better when trained on very little datasets, but that neural networks produce better predictions when both more data is available and models are used to produce more than one output in order to account for the intrinsic ambiguity of some of the language pairs. In preliminary experiments on real data, we observed that, on small and noisy datasets, the statistical model performs consistently better than the neural model, but that the neural model seems to have learned higher level morphological information. Further experiments need to be done, both with less noisy, bigger real datasets (e.g. manually curated) and with more complex artificial data, with more sound changes and added noise separating the proto-language from its daughter languages.

7. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Cser, A. (2016). *Aspects of the phonology and morphology of Classical Latin*. Ph.D. thesis, Pázmány Péter Katolikus Egyetem.

Dekker, P. (2018). Reconstructing language ancestry by performing word prediction with neural networks. *Master. Amsterdam: University of Amsterdam*.

Duddington, J. (2007-2015). espeak text to speech. `http://espeak.sourceforge.net/index.html`.

Fourrier, C. and Sagot, B. (2020). Methodological Aspects of Developing and Managing an Etymological Lexical Resource: Introducing EtymDB-2.0. In *Twelfth International Conference on Language Resources and Evaluation (LREC 2018)*, Marseilles, France. (to appear).

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

Inkpen, D., Frunza, O., and Kondrak, G. (2005). Automatic identification of cognates and false friends in french and english. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, volume 9, pages 251–257.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

List, J.-M., Greenhill, S. J., and Gray, R. D. (2017). The potential of automatic word comparison for historical linguistics. *PLOS ONE*, 12(1):1–18, 01.

List, J.-M., Walworth, M., Greenhill, S. J., Tresoldi, T., and Forkel, R. (2018). Sequence comparison in computational historical linguistics. *Journal of Language Evolution*, 3(2):130–144, 07.

List, J.-M. (2019). Automatic inference of sound correspondence patterns across multiple languages. *Computational Linguistics*, 45(1):137–161.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Mann, G. S. and Yarowsky, D. (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*, NAACL ’01, pages 1–8, Pittsburgh, Pennsylvania. Association for Computational Linguistics.

Mulloni, A. (2007). Automatic prediction of cognate orthography using support vector machines. In *Proceedings of the ACL 2007 Student Research Workshop*, pages 25–30, Prague, Czech Republic, June. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Osthoff, H. and Brugmann, K. (1878). *Morphologische Untersuchungen auf dem Gebiete der indogermanischen Sprachen*. Number 1 in Morphologische Untersuchungen auf dem Gebiete der indogermanischen Sprachen. Hirzel.

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

[16] Since the algorithms have to predict n different answers for n-best prediction (when only one answer might be correct), it is expected that in each set of predicted words, some will be nonsensical; we present here words ordered by algorithm confidence.