11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Homograph disambiguation with contextual word embeddings for TTS systems

Marco Nicolis, Viacheslav Klimkov

Amazon

nicolism@amazon.com, vklimkov@amazon.com

Abstract

We describe a heterophone homograph (simply 'homograph' henceforth) disambiguation system based on per-case classifiers, trained on a small amount of labelled data. These classifiers use contextual word embeddings as input features and achieve state-of-the-art accuracy of 0.991 on the English homographs on a publicly available dataset, without any additional rule system being necessary. We show that as little as 100 sentences are sufficient to train a light-weight dedicated classifier, provided the dataset is sufficiently balanced, i.e. all versions of the homograph are adequately represented. We further add data in cases where the original dataset is deeply unbalanced (i.e. one homograph version overwhelmingly represented). This is effectively a special case of active learning, by which we select additional cases of the under-represented homograph versions and show an 11% relative improvement for such cases. We finally provide a solution to drastically reduce the size of our models, via sparsification.

Index Terms: homograph disambiguation, TTS, front-end

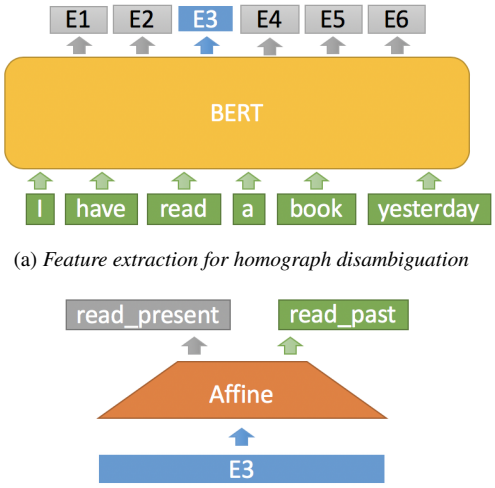
1. Introduction

A homograph is a 'word that is spelled the same as another [...] but which differs in sound *and* meaning, such as *tear* (to separate or pull apart) and *tear* (a secretion from the eye)' ([1, p. 3]). The correct classification of homographs is a long-standing issue affecting the text analysis component of text-to-speech (TTS) systems. While homographs are present in many of the world's languages, this paper focuses on American English only. For the sake of reproducibility we report the performance of the proposed approach on the openly available dataset from [2][1]. We additionally use a small internal dataset to augment the main dataset in cases where one of the two possible pronunciations of a given homograph is severely under-represented in the main dataset. We show this brings about a large improvement for these cases.

All approaches to homograph disambiguation agree that correctly interpreting the context surrounding the homograph word is key to their disambiguation. Different types of linguistic information are, however, crucial in different cases. The lexical semantic context allows the disambiguation of *lexical* homographs like *bass* (fish vs. musical instrument, both nouns), whereas morphosyntactic constraints regulate the *morphosyntactic* class, which can generally be disambiguated via POS tags (e.g. nominal vs. verbal *impact*).

Rule based and supervised machine learning systems have historically directly encoded the multiple sources of homography in the features used for disambiguation. The algorithm described in [3] heavily depends on collocations and their statistical distribution. For example, the presence of semantically related words within a certain distance from the homograph word (e.g. does the word *water* occur in the ± 20 word window around the target word *bass*?) or the co-occurrence of certain word sequences (e.g. the [*of* lead *in*] sequence correlates with the [lɛd] pronunciation of *lead*, whereas the [*the* lead *in*] sequence correlates with the [lɪd] pronunciation). Twenty years later, [2] feed a multinomial log-linear model (one for each homograph) with similar types of features: word context features (left and right bigrams around the homograph), POS tag and capitalization (*Polish* vs. *polish*).

Transformer-based contextual word embeddings (CWE), i.e. the BERT family ([4], [5], [6], etc.) appear to be ideally suited for the purpose of homograph disambiguation. The vectors they produce are inherently context-sensitive: the vector for *bass* will be different depending on the surrounding context and the information they encode spans across the whole traditonal NLP pipeline, including the lexical semantics and morphosyntactic information needed to disambiguate homographs. See ([7] and [8] for a recent overview). It follows that the adoption of CWEs allows for a unified system encompassing all homographs, be their disambiguation based on morphosyntactic or lexical semantic factors. In the remainder of this paper we show that this approach yields SOTA accuracy; more importantly it does not require complex, expensive to produce and hard to maintain rule-based components.



(a) *Feature extraction for homograph disambiguation*

(b) *Homograph disambiguation with simple classifier*

Figure 1: *Homograph disambiguation pipeline*

2. Model description

Contextual word embeddings (CWEs) provide numerical representation of the word in the given context. They are trained on large text corpora on the masked language modeling task (i.e.

[1] https://github.com/google/WikipediaHomographData

10.21437/SSW.2021-39

prediction of masked words in a sentence). With reproducibility in mind, in this work we use the BERT pretrained model[2] distributed for MXNet ([4]), one of the most widely adopted models openly available. CWE models are resource hungry, which makes makes them not always suitable for a production environment. Thus, we also experimented with the ALBERT model, a light-weight CWE model ([5]). We use the pretrained model[3] distributed for Pytorch.

We only utilize the relevant embedding for the token of interest (homograph), under the assumption that sufficient relevant contextual information is expressed in it. This is depicted on Figure 1. The extracted embedding is then fed to a simple per-homograph classifier. In our experiments we used a logistic regression classifier, trained on pairs of contextual embeddings and correspondent homograph cases. For example, for the homograph "*read*", the training set contains 52 and 60 sentences with past and present form of it respectively.

The per-homograph logistic regression model is trained using the MXNet framework. More complex models, such as multi-layer perceptron, were explored, but did not prove to be beneficial. We also prefer to present the simplest approach, as this makes it easier to reproduce, and it highlights the usefulness of the underlying CWEs for the homograph disambiguation task. The model is trained using stochastic gradient descent using Adam optimizer with a learning rate of $1e-3$. The models were optimized on all training examples for the given word simultaneously. To compensate for the small amount of training data and excessive representation capabilities of input features, it proved beneficial to apply *l2*-regularization with weight 0.01.

Each classifier is a matrix of size (*embedding_dimension* x *homograph_cases_num*), where *embedding_dimension* is either 1024 or 768 depending on the version of the BERT model used, and *homograph_cases_num* is typically 2, corresponding the two pronunciations of a given homograph. That results in 3kB per model or 0.5MB for all 162 homographs from the dataset in case of quantization to float16.

3. Experiments

[2] provide an overview of Google's homograph disambiguation system. The baseline model they describe as Google's production model at the time of writing is rule based, while the paper introduces an ML approach where the features described in section 1 (positional features, capitalizaton, POS tag) are fed to a multinomial log-linear model. Combining rules and ML model yields an important boost in accuracy (see Table 1). We use the results in [2] as our baseline, and report results of our model trained on identical data (Experiment 1) and augmented data (Experiment 2). We follow [2] in reporting both micro accuracy (percentage of correctly classified examples) and macro accuracy (arithmetic mean of per-homograph accuracies) for all our models.

3.1. Experiment 1: Data from [2]

One of the stumbling blocks of systems relying on POS tags for disambiguation concerns the taggers' accuracy for non-trivial cases, for example cases where nouns and verbs share the same orthography (e.g. verbal and nominal *impact*, TIPA). While the accuracy of SOTA taggers is at almost 98%[4] on Penn Treebank data, [9] show that the accuracy of top parsers ranges between 57% and 74% on their crowdsourced dataset specifically targeting ambiguous nouns and verbs. In that paper, trivial cases were filtered out from the dataset[5]: for example, given a N/V ambiguous word (e.g. *impact*), both the [Det + Word] and [Aux + Word] case will provide trivial and exceptionless disambiguation evidence (N for Det + Word: *the impact* and verbal for Aux + Word *will impact*). [9] use the extra data from their challenge dataset to retrain the tagger in [11]; they in addition add ELMo embeddings [10] to their pipeline. The overall absolute improvement over the [11] baseline was 14% (from 75% to 89%); adding only ELMo yielded an improvement of 7.2%, while only adding the new training data improved the result by 10%. In the same paper, the authors measure the improvement achieved by their tagger on the dataset from the [2] paper. The improvement over the ML model in [2] was 1.3% when adding ELMo embeedings to the tagger, 0.3% when adding the challenge dataset as training data, while the combined effect Elmo+dataset was again an improvement of 1.3% (see table 1). This result suggests that CWEs can successfully be used to improve parsers' performance, and thus indirectly improve homograph disambigutation.

Table 1 reports the reults of the baselines described in [2] as well as the accuracy obtained by [9] on the same dataset. As mentioned above, the source of improvement introduced in that paper is twofold: training a POS tagger with additional 'challenging' data including Noun/Verb homographs (e.g. *impact*), and adding ELMo [10] embeddings to the pipeline. Virtually all of the reported improvement on the homograph disambiguation task stemmed from adding ELMo. This result strongly suggests that the use of CWEs is very beneficial for the disambiguation of homographs.

For our first experiment we created one model per each of the 138 homographs listed in [2] and fed the model with CWEs obtained from BERT and ALBERT pre-trained models of varying size. Both the training and test data were exactly the same as those used in [2]: for each homograph there are about 90 sentences in the training data and 10 in the test data. We did exclude the homograph *conglomerate* from our analysis, since there are no cases of verbal *conglomerate* in the training data, while there is one instance in the test data[6].

The results indicate that all the models we trained outperform both the ML model of [2] and the POS-tagger based solution in [9]; in the case of ALBERT-base embeddings, the result obtained outperforms both the ML and ML + Rules models reported in [2]. Models with BERT and ALBERT embeddings perform quite similarly to each other on this task. We leave a more careful analysis of the differences to future work.

Following a reviewer's suggestion, we carried out Leave-One-Out Cross-Validation for our BERT base model, given the very small size of our dataset. The results are very similar to those reported for the original split (98.5% accuracy with cross-validation vs 98.8 accuracy for original split).

For most of the remainder of the paper we will focus on BERT large, as we are independently carrying out additional

[2] https://pypi.org/project/BERT-embedding/

[3] https://huggingface.co/transformers/model_doc/albert.html

[4] http://nlpprogress.com/english/part-of-speech_tagging.html

[5] urlhttp://goo.gl/language/noun-verb

[6] This affects our numbers minimally: 10/10 cases in the training data feature the nominal variant, and so do 9/10 cases in the test set. We would thus have scored 9/10 on this homograph; however, given the verbal variant is not represented in the training data, it is simply impossible to learn anything about it, and we thus feel excluding this homograph is the correct solution

	Experiment 1	
System	Micro %	Macro %
[2] Rules	89.0	88.6
[2] ML	95.4	95.1
[2] ML + Rules	99.0	99.0
[9] POS + ELMo	96.7	96.7
Ours + BERT base	98.8	98.8
Ours + BERT large	98.7	98.7
Ours + ALBERT base	**99.1**	**99.1**
Ours + ALBERT large	98.3	98.3
Ours + ALBERT x-large	97.6	97.6
Ours + ALBERT xx-large	98.6	98.6
Ours + BERT base (leave-one-out)	98.5	98.5

Table 1: *Our models' accuracies vs. baselines for Exp. 1*

work on this model and the accuracy difference between this model and other BERT variants is rather small.

We conclude this section by observing that models appear to have learned significant cues from the surrounding context, beyond what provided by the immediately surrounding words. We for example get the correct result for the two sentences considered challenging in [2] ((1), (2) in Table 2). The challenge in sentence (1) is the sequence '*wind up*', which is frequently associated with a phrasal verb interpretation. In sentence (2) the sequence '*to present*' frequently appears as an infinitival verb. For both sentences our models assign almost 100% probability to the correct tagging. We report the implied probabilities as $softmax(logits)$ in table 2. As some sentences are genuinely ambiguous, we checked whether our models assign lower probabilities to such cases. This is indeed the case. Sentence (3) is ambiguous, as the verb *read* can be interpreted as present or past tense depending on the surrounding context. A survey conducted with 10 English native speakers confirms that the preferred interpretation of *read* in (3), when uttered in isolation, is overwhelmingly with the verb in the past tense: this ultimately has to do with English quite rigidly mandating the present progressive form to express continuous/progressive aspect (contrary to, say, Romance languages where present tense can be used to express continuous aspect). Our models correctly assign a mild preference to the VBD interpretation.

Sentence	% [Tag]	% [Tag]
(1) There may be **winds** up to 20 miles per hour	> 99.9 [N]	< 0.01 [V]
(2) Smith has played Trophy matches for the county from 1993 to **present**	> 99.9 [N]	< 0.01 [V]
(3) I **read** a book	63.52 [VBD]	36.48 [VB]

Table 2: *Challenging sentences and implied probability (data for model trained on BERT large embeddings)*

3.2. Experiment 2: Data augmentation for highly unbalanced train sets

The 19 homographs for which we do not obtain 100% accuracy on the test set (10 sentences) for the BERT large models are listed in Table 3.

Table 3 shows that 10 out of 19 homographs for which accuracy is less than 10/10 are homographs whose training set is deeply unbalanced, containing less than 10 cases for one of the two versions of the homograph word (column 'Split' in table 3). We selected 9 of the 10 cases (indicated by the light shading in the table) and manually selected additional training sentences from an internal dataset made up mostly by fiction materials. We added 50 additional sentences for each of these homographs, ensuring that we would include at least 10 sentences for the 'weak', under-represented version of the homograph.[7]. It is worth reiterating that the annotator selecting the additional sentences was not aware of the results obtained by our model, and thus the selection was completely unbiased (i.e. there was no overt attempt to fix the incorrect cases). The new splits obtained are included in the NewSplit column of table 3. A reviewer correctly points out that the numbers in the 'Split' column in table 1 does not always add up to 90. This is an issue with the original dataset described in [2], which we are adopting in this paper. In several cases, the train data does not contain 90 sentences, but fewer.

	Experiment 2 data			
Word	Split	OrigAcc	NewSplit	NewAcc
approximate	78/11	9/10		
compress	84/4	9/10	109/29	10/10
confines	73/16	9/10		
content	1/89	9/10		
correlate	7/83	8/10	17/123	10/10
duplicate	67/23	9/10		
escort	81/9	9/10	109/31	9/10
graduate	87/3	8/10	127/13	9/10
incline	86/3	9/10	126/13	10/10
increment	78/11	9/10		
insert	47/43	9/10		
intrigue	86/4	9/10	126/14	10/10
invalid	67/22	9/10		
invert	27/62	9/10		
laminate	87/3	9/10	126/14	10/10
pasty	63/22	8/9		
perfume	88/2	9/10	134/6	10/10
upset	65/24	9/10		
transplant	85/5	9/10	126/14	10/10
Total	**79/90**	**87.8%**	**88/90**	**97.8%**

Table 3: *Split, old and new accuracy for homographs with* $< 100\%$ *in Experiment 1*

	Experiment 2	
System	Micro %	Macro %
Ours + BERT large	**99.3**	**99.2**

Table 4: *Our model's overall accuracy for Exp. 2*

3.2.1. Results

We re-trained our classifier fed with BERT large embeddings with the new added materials described in section 3.2, in the same way as in Experiment 1. We report the new accuracy obtained in the NewAcc column in Table 3. For the nine homographs targeted the accuracy improved from 87.8% to 97.8%. Adding a small amount of extra labelled data to very unbalanced cases thus yielded an absolute improvement of 10% and a relative improvement of 11.38%.

[7] In cases for which we couldn't find 10 relevant cases in the first 1000 lines of the relevant corpus, we added however many 'weak' cases we found up to that point. We did not consider the homograph *content* because all 1000 cases at the top of our corpus featured nominal *content*

The overall accuracy of the our system built using BERT large embeddings improved to 99.3% and 99.2% (for micro and macro accuracy respectively), as shown in Table 4.

These results confirm that making the training set less unbalanced improves accuracy significantly. As the number of sentences added was just ten per target word, a new approach to fixing bugs related to homographs emerges: while relying on hand-crafted rules requires careful and skilled labour, attempting to fix a bug via data augmentation simply requires a native speaker of the language to select a handful of sentences matching the desired version of the homograph.

3.3. Experiment 3: Gradual sparsification of affine transform weights

Any online TTS system benefits from low latency. Experiment 3 investigates whether the size of our models can be reduced, thus improving the overall latency of our system. We aim at reducing the size of the model by applying gradual sparsification of affine transform weights during training.

The amount of non-zero weights was decreased throughout the training to 10% of the original amount, masking weights with the lowest amplitude every 100 training steps. This yielded a reduction of the size of the classifier for each homograph to 300 bytes, with minimal effect on the accuracy of the resulting model. Explicit sparsification required to decrease weight of *l2*-regularization to 0.001.

We applied sparsification to two models from Experiment 1: the models built on BERT base and ALBERT base. The results are displayed in 5.

	Experiment 3	
System	Micro %	Macro %
Ours + BERT base	98.8	98.8
Ours + BERT base (sparse)	98.8	98.8
Ours + ALBERT base	99.1	99.1
Ours + ALBERT base (sparse)	98.8	98.8

Table 5: *BERT base and ALBERT base with and without sparsification*

The results of this experiment show that sparsification introduces no accuracy loss in the case of BERT base, and a mild accuracy decrease (from 99.1% to 98.8%) in the case of ALBERT base. The size of the models obtained by way of sparsification is about 10 times smaller than the models described above for Experiment 1 (from 3kb per model to 300 bytes per model).

Model sparsification is thus a viable solution for online TTS systems where model size and latency are a concern.

4. Conclusions

This paper introduces a fully ML-based homograph disambiguation system based on contextual word embeddings. The proposed approach achieves SOTA results, without the need of any *ad-hoc* rules. This paves the way a fully automated approach to homograph disambiguation for TTS systems: the need for expensive, language-specific and hard to maintain rule systems becomes much less central. While this paper has only dealt with English, the availability of CWEs in many languages suggests that the simple approach described can be successfully extended cross-linguistically. We have further shown that balanced train datasets are crucial, as adding a few examples of the under-represented variant in unbalanced cases improved our accuracy by over 11% (relative). The method proposed, although performed manually, is effectively a special case of active-learning: relevant data is selected to increase the number of under-represented variants. Throughout the paper we keep in mind integration of proposed homograph disambiguation system into production environment: using light-weight ALBERT features, using pretrained CWE features that can be reused for other purposes within TTS framework, using sparsified per-homograph models.

5. References

[1] J. Hobbs, *Homophones and Homographs: An American Dictionary, 4th ed.* McFarland, Incorporated, Publishers, 2006. [Online]. Available: https://books.google.co.uk/books?id=vCUTBQAAQBAJ

[2] K. Gorman, G. Mazovetskiy, and V. Nikolaev, "Improving homograph disambiguation with supervised machine learning," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. Miyazaki, Japan: European Languages Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1215

[3] D. Yarowsky, "Homograph Disambiguation in Text-to-Speech Synthesis," in *Progress in Speech Synthesis*, J. P. H. van Santen, J. P. Olive, R. W. Sproat, and J. Hirschberg, Eds. New York, NY: Springer New York, 1997, pp. 157–172. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-1894-4_12

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *arXiv:1909.11942 [cs]*, Feb. 2020, arXiv: 1909.11942. [Online]. Available: http://arxiv.org/abs/1909.11942

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692 [cs]*, Jul. 2019, arXiv: 1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692

[7] I. Tenney, D. Das, and E. Pavlick, "BERT Rediscovers the Classical NLP Pipeline," *arXiv:1905.05950 [cs]*, Aug. 2019, arXiv: 1905.05950. [Online]. Available: http://arxiv.org/abs/1905.05950

[8] C. D. Manning, K. Clark, J. Hewitt, U. Khandelwal, and O. Levy, "Emergent linguistic structure in artificial neural networks trained by self-supervision," *Proceedings of the National Academy of Sciences*, 2020. [Online]. Available: https://www.pnas.org/content/early/2020/06/02/1907367117

[9] A. Elkahky, K. Webster, D. Andor, and E. Pitler, "A challenge set and methods for noun-verb ambiguity," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2562–2572. [Online]. Available: https://www.aclweb.org/anthology/D18-1277

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[11] B. Bohnet, R. T. McDonald, G. Simões, D. Andor, E. Pitler, and J. Maynez, "Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings," *CoRR*, vol. abs/1805.08237, 2018. [Online]. Available: http://arxiv.org/abs/1805.08237