Homonym normalisation by word sense clustering: a case in Japanese

Yo Sato
Satoama Language Services
Kingson Upon Thames, U.K.
yo@satoama.co.uk

Kevin Heffernan
Kwansei Gakuin University
Sanda, Hyogo, Japan
kevin@kwansei.ac.jp

Abstract

This work presents a method of word sense clustering that differentiates homonyms and merge homophones, taking Japanese as an example, where orthographical variation causes problem for language processing. It uses contextualised embeddings (BERT) to cluster tokens into distinct sense groups, and we use these groups to normalise synonymous instances to a single representative form. We see the benefit of this normalisation in language model, as well as in transliteration.

1 Introduction

This work presents a method of clustering homonyms that uses contextualised word embeddings, assesses its performance for polysemy detection and shows two use cases in Japanese, normalisation and transliteration. Japanese has multi-typed orthography, using three different alphabets, which creates challenges as homonyms/homophones are extensively observed. Our main proposal is to detect semantically equivalent groups in them, thereby cerating normalised, semantically less noisy data in pre-processing, to the benefit of language modelling. We also apply the same technique to a long-standing 'annoyance' in Japanese input, i.e. transliteration process.

Homonymy is a universal phenomenon, and distinguishing their distinct meanings is a major problem in language processing. The advent of contextualised word embeddings (Peters et al., 2018; Devlin et al., 2018) has made it possible to differentiate token level occurrences, and they have indeed been utilised for supervised word sense disambiguation (Huang et al., 2019), but whether it can also capture latent meaning distinctions without labels is unclear, especially when the meaning boundaries are vague (Mickus et al., 2020).

Our strategy is to detect relatively clear-cut cases by using a *halting criterion* on the number of clusters. Japanese offers a unique platform to conduct such clustering on, since along with homonyms, their opposite counterpart, *heterographs* —words semantically and phonemically identical but written differently— systematically coexist. We will show first that accurate enough clustering is possible via contextualised embeddings and then that the clustering improves the two aspects mentioned above, normalisation and transliteration.

2 Terminology

Terminology in which to describe homonymy and heterography and related phenomena is rather confusing and often confused, and some tidying-up is in order for clarity. We have at least six related terms, homo-/hetero- prefixes with -phone/-nym/-graph suffixes that describe a relation that holds in a set (mostly pair) of word types. In accordance with how the terms are conventionally used, we could understand these roughly to correspond to the combinations of the identity/difference in three parameters, sound, writing and meaning: the combination of same sound, same spelling but different meaning is for homonyms and so on. Leaving out the

Proceedings of the 28th International Conference on Computational Linguistics, pages 3324–3332
Barcelona, Spain (Online), December 8-13, 2020

uninteresting combinations of the same and different in all these respects, we are left with six combinations. The following table shows the correspondence.

Sound	**Writing**	**Meaning**	Term	Examples Eng	Examples Jp
same	same	diff	homonym	bank/bank	くま (熊)/くま (隈)
same	diff	diff	homophone	there/their	熊/隈
diff	same	same	homograph	either [iːðə]/[ɑɪðə]	七 [nana]/[ʃɪʧɪ]
diff	same	diff	heteronym/heterophone	bow/bow	?
diff	diff	same	synonym	stop/halt	停める/停止する
same	diff	same	heterograph	racket/racquet	いも/芋

Table 1: Categorisation of homo-/hetro- terms

While the conventional usage unfortunately does not lend itself to a clear one-to-one mapping, our targets are fortunately amongst the teminologically clear ones, namely homonyms/homophones on one hand, and heterographs on the other. That is, words pronounced identically but not synonymous, and ones that are identical in meaning but are written differently. Homonyms and homophones are differentiated based upon whether the writing is also identical (homonyms) or not (homophones) in addition to sound, but this distinction is sometimes not essential, and we take liberty to use the term homonym to represent both, unless the distinction is crucial. In short, we are dealing with one-to-many mappings in both directions, from sound to meaning, and from meaning to sound. It is important to note, in this relation, an inter-language contrast, mainly caused by the orthographical charateristics, i.e. the phonemic/ideographic contrast. Heterography is not a prominent phenomenon in a single-alphabet orthography, but is pervasive in Japanese. With both heterography and homonymy pervasive, Japanese poses the issue of multiplicity in both directions, calling for resolution on both counts. Clustering of word types provide a uniform solution for these problems.

3 Motivation

With our terminology definition in place we can now succinctly state what our objectives are. Our ovearching goal, homonym normalisation, is to render the data more compact (less type-sparse) and less ambiguous by grouping together heterographs —different-looking synonymous words— and separating out homonyms —same-looking heteronymous words. We do this by clustering, attempting to achieve the following: heterographs be merged into a single group and homonyms be grouped into different meanings.

Aside from being free from labelling, clustering has this advantage of achieving these two tasks in one go, which would have to be done in two separate tasks in the classification approach. There is a further advantage in the clustering approach for a set of sutble cases: avoiding overfitting on words with very small semantic differences, i.e. 'very close' synonyms such as 'midday' and 'noon'. Furthermore, Japanese has a sometimes quite extensive set of 'almost synonymous' homophones, due to the 'borrowing' of Chinese characters with similar meanings that are riginally pronounced differerntly in Chinese but end up as homophonic, in the absence of differentiable equivalents in Japanese. Such cases are observed particularly for verbs and adjectives. For example, homophonic '停める', '止める', '留める', '駐める', all roughly meaning *to stop/stall* are so close that the choice could be simply up to the stylistic preference of the writer. We would like to allow these close synonyms to belong to the same group. Another advantage is a robust, gradient approach to vagueness. A homonym can show 'shades' of meanings. 'Bank' in bank account and food bank may be considered a borderline case, as opposed to river bank vs. bank account. Clustering approach provides the flexibility to handle all these cases.

Generally speaking, contexts does the job of disambiguation, even a very short one at times. For example, for Japanese, くま (*kuma*), homonymous between *bear* (animal) and *black eye*, can be disambiguated by a single-verb context: くまが出た/くまができた ('a *kuma* showed up/developed'). So our hope is that with the use of contextual embeddings and an appropriate halting criterion, we will end up with two clear cut clusters. For those borderline cases like the above, on the other hand, we let the data decide if the purported synonyms are contextually similar enough.

In the context of disambiguation, the multiplicity of alphabets in Japanese is something of a mixed blessing. The co-existence of phonemic alphabets (two sets of kana) and ideographic characters (kanji) does make the task of clustering more complicated, but the latter provides a natural reference for homonymy. Sticking with our *kuma* example, of which we earlier had the hiragana version, it can be written in katakana (クマ) too, where we have exactly a parallel situation (homonymic and two-way ambiguomus), and then we have ideographic versions, 熊 and 隈, which do correspond to distinct meanings. Our desired end results would therefore be to end up with two clusters for the whole set, 'bear's and 'black-eye's, the first containing 熊 and bear-meaning kana occurrences, the second 隈 and black-eye meaning kana occurrences.

Another important motivation comes from a practical application essential for the input of ideographic orthography on computer: input method, or transliteration. In Japanese or Chinese input, the user copes with the numerous characters by inputting first phonemic form (kana for Japanese, pinyin for Chinese) and 'convert' it into ideographs. This conversion process itself is cumbersome enough, but in the presence of homophones, the user is further required to *choose* amongst multiple ideographs shown on screen, even when the the choice is obvious from the context. This issue can also be handled by our clustering model, by additionally turning the model into a predictor. We will see this application later in Section 6.2 too.

4 Related work

Clustering on word sense has attracted interest in the community of computational semantics since early days, with the unsupervised word sense induction tasks often featuring in SemEval workshops (Agirre and Soroa, 2007; Manandhar et al., 2010). Theoretical motivations for word sense clustering are well described in (McCarthy et al., 2016), as well as the comparison with the traditional classification method. Since word embeddings emerged as a standard tool in NLP, and particularly after the advent of contextualised embeddings, embeddings-oriented work also started to appear (Goyal and Hovy, 2014; Amrami and Goldberg, 2019) for polysemy detection, while others have started to use them for diverse applications such as transfer learning (Ustalov et al., 2018), paraphrasing (Cocos and Callison-Burch, 2016) and search engine improvement (Kutuzov and Kuzmenko, 2016).

Normalisation has been conducted primarily on heterographs (such as abberation from the standard form) with a classification method (Han et al., 2011), including Japanese (Saito et al., 2014), with noisy text as the target. Contextualised embeddings have started being used in this context (Muller et al., 2019), again usually under the supervised framework. Work combining normalisation with clustering is not so common though exceptions exist (Zalmout et al., 2019).

As for transliteration systems into ideographs, several statistical methods have been proposed and implemented both in Chinese and Japanese. For Chinese, neural-net based systems are actively developed, such as (Huang and Zhao, 2018) and (Huang et al., 2018). For Japanese, Google's Mozc is probably the most popular statistical input method commercially available (Kudo et al., 2011). Being essentially a bigram model, however, it is not sensitive to meaning distinctions of homophones when the relevant context is located farther than the immediate vicinity, or the right choice is infrequent. For example, consider the plausible rendering of *kōsō* in this context: 仏教史に名を残した中国の _____ ('a Chinese _____ who has gone down in the Buddhist history'). For the human mind, the clear choice, though its relative infrequency, is 高僧 ('prominent monk') rather than other homophones, such as 高層 ('high-rise'), 構想 ('design') and 抗争 ('infighting'), which are nevertheless its top candidates. More advanced engines have been proposed, e.g. a recurrent neural-network (RNN) based model by (Okuno, 2016), but it is stated that the latency issue prevents such a model from being deployed. While we do not mean to construct a full input method, we will show that our 'module' can improve the choice in many cases without much overhead.

5 Method

5.1 (Re-)Training word embeddings

For a clustering on contextual meaning like ours, word embeddings are a natural choice as data representation. However, as we cluster the token occurences of homonyms —on the phonemic level these constitute a single type— these tokens need to be encoded in different embeddings according to their context. The classical, word-type based word embeddings, such as word2vec (Mikolov et al., 2013), will therefore not work as they are, though it is possible to use the mean vector of the embeddings of the surrounding 'window'. A better alternative is a modern *contexturalised* variety such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018).

We will mainly use BERT as it performs best. Based on the pre-trained Japanese model trained on Wikipedia (Kikuta, 2019), fine-tuning was conducted using two corpora, the National Institute of Japanese Language and Linguistics's Corpus of Spontaneous Japanese, (National Institute of Japanese Language and Linguistics, 2003) and Mainichi Shimbun Corpus (Mainichi Shimbun, 1995), or to be precise 90% of each, since the rest will be used for evaluation. To align with the way Kikuta (2019) pre-processes the data, the data has been tokenised with MeCab (Kudo et al., 2004) and SentencePiece (Kudo and Richardson, 2018) before fine-tuning. MeCab is not just a tokeniser but a 'morphonological analyser' which produces information such as PoS and pronunciation amongst others, which can be utilised for other pre-processing purposes. With the realisation that there are not enough hiragana and katakana instances either in the original training data and our additional corpora, we artificially converted 10% of the kanji occurrences into hiragana and katakana (5% each), using the pronunciation information in the MeCab output. For a comparison purpose, we also extracted the word2vec CBoW embeddings from the same corpora and the same tokenisation and computed the means of the window of 5 (each side).

5.2 Clustering

Our clustering quality depends on two crucial factors: the membership and the number of produced clusters. The membership quality then mainly depends on that of data coherence and clustering algorithm, and the number on the 'halting criterion' of clustering.

We first extracted sets of embeddings that correspond to homonyms and heterographs, that is the token instances with identical pronunciations, and clustering is conducted on each set. To avoid cases without much context, we set the threshold sentence length to 5. For clustering, whereas a variety of algorithms are available, we chose Gaussian Mixture Model (GMM) on account of its relative simplicity and good performance. GMM also has an advantage of being capable of funcioning as a predictor, which will be used for transliteration and evaluation.

We also have a number of choices for halting criteria. We again here take a practical approach and choose the best performing, or best suited, one to this particular task: Gap Index (Tibshirani et al., 2001).

6 Applications

To gauge the benefit of homonym normalisation by clustering, we applied the results to two applications, language model and transliteration. We will evaluate our clustering performance on the normalisation results themselves, but also on these two applications.

6.1 Normalised language model

One of the likely benefits of normalisation in general is compact and improved language modelling, as the originally fragmented tokens are unified to alleviate sparcity. Therefore we use the clustered (normalised) data to build a language model and see how much better its performance becomes in comparison with the model trained on the original un-normalised data.

After clustering, we give the same form to the instances of heterographs belonging to the same cluster, while assigning different ones to the homonyms belonging to different clusters. The tokens in a set of homonyms and heterographs are first normalised using the hiragana form, and then indexed with a cluster number, word-in-kana$_1$, word-in-kana$_2$ and so on. For example in our pet example of *kuma*, supposing the (successful) clustering into two groups, all the tokens whose surface forms are one of くま (hiragana), クマ (katakana), 熊 (kanji meaning bear) or 隈 (kanji meaning black eye), will be rendered either くま$_1$ or くま$_2$, say the first meaning bear and the second black eye.

We built language models of two flavours, a simple N-gram model and an RNN model, in line with (Okuno and Mori, 2012), which includes a Viterbi decoder for segmentation and copes with the input of multiple representations. The results of the comparison between normalised and unnormalised models will be shown in Section 7.2.

6.2 Transliteration module

We will also apply the clustering model to homophone resolution in the context of transliteration, popularly known as *kana-kanji conversion*. Generally, the conversion task itself consists of, given a sentence written all in phonemic kana form without word segmentation, transliterating ideograph-renderable constituent words into ideographs.

Transliteration however presupposes word-segmentation. The initial input is in phonemic kana without spaces, and this input needs to be segmented into words for any potential homophones to be detected. Once this preliminary task is done, homophone candidates can be identified, at which point our cluster predictor can kick in to identify the right group of meaning.

For word segmentation, we employ a simple N-gram model (included in (Okuno, 2016) as a baseline). When a homophone candidate is detected, our clustering-based resolution module is triggered. The module first finds the BERT embedding for that token instance, and our GMM predictor will then return the cluster it belongs to.

In each cluster in our model, one or two dominant, usually ideographic, form should be present which represent the meaning the cluster is associated with. Thus we could use the most frequent item as its 'label', and hence the conversion target. Therefore, when the cluster is returned, so is the conversion target. Incidentally the target can, despite the name 'conversion', be in the kana form if it is the most frequent one.

7 Evaluation

7.1 Evaluation of clustering

7.1.1 Gold sets

We evaluate the resulting clusters against the 'gold standard', created from the portion of our two corpora reserved for testing, by manually rendering into hiragana indexing the homonyms and heterographs in the phone-shared token set in the scheme described in Section 6.1. Since we are concerned about not just differentiating tokens but sometimes merging heterographic items when their meanings are close, we could put some kanji-rendered homophones into a single group with the same index. Following is part of an example set, for homophones and heterographs for *seisaku.*

制作 者の意図は少しわかりづらいと言わざるを得ない。→ せいさく$_1$
(The intention of the production team is rather hard to interpret.)
この図面の製作 にあたっては多くの技術者が関わった。→ せいさく$_2$
(Many engineers were involved in producing of this plan)
政策 立案を行ったのは通産省のエリート官僚たち。→ せいさく$_2$
(It is the elite buraucrats who designed the policy)
せいさく の是非を問う議決が行われた。→ せいさく$_2$
(The vote was taken to decide whether to enforce the policy)
...

Notice two homophones, 制作 and 製作, are grouped with the same index as they are considered semanitcally similar. While kana-written homonyms may be rare for some sets, we have artificially converted some homonyms into kana, the last example exemplifies one such case.

We also set a threshold of a sentence length at 5 words, following the training set. The sets number 245, and the mean reference cluster size for goldsets, thus obtained, is approximately 4.2.

7.1.2 Metrics and results

Evaluation of clustering performance is somewhat less clear-cut than that of classification, in the sense that what level is considered ‘good’ and ‘bad’ depends on the task and is somewhat subjective. We use two of the relatively established metrics, Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and V-measure (Rosenberg and Hirschberg, 2007) on the clustering results on CBoW mean embeddings and BERT embeddings. The upper bound for both metrics is 1, while the lower bounds are -1 and 0 respectively. In both metrics, close to zero or less than zero indicates randomness, and hence at the very least, both results are well over the random expectation, though the CBoW figures can be said to be in the range generally regarded as poor.

Meanwhile BERT registers quite satisfactory performance. V-measure is not so good as ARI, but this can be related to the characteristics of each metric, as the former is considered to be a good measure for skewed clusters, whereas the latter for balanced ones (Romano et al., 2016). We generally have both types of cases, but on the whole are on the ‘balanced’ side (the mean skewness at -0.28). Furthermore, we tend to have a better completeness than homogeneity, the two components of V-measure, where homogeneity penalises over-split. This can therefore be related to the halting criterion, and implies that the membership itself is better preserved.

Model	**ARI**	**V-measure**
		(Homogeneity/Completeness)
BERT	.71	.51 (.41/.69)
CBoW	.29	.31 (.24/.45)

Table 2: Clustering of homophones into senses

7.2 Normalised language model

In Table 3, we show the comparison between normalised and non-normalised data for two models, N-gram and RNN, on their perplexity and prediction accuracy (w/o: without normalisation, with: with normalisation). We have to be mindful in this comparison of the difference in the vocabulary size. As a result of clustering, we will naturally have a reduced vocabulary, and prediction will be easier with a smaller vocabulary. Therefore for a fair comparison, we also show, for the without-normalisation figures, the *adjusted* versions of accuracy and perplexity too, where the original figure are divided by the inverse of the reduction rate of, the whole vocabulary for perplexity, and of the choice in homonyms for prediction.

Model	**Perpl./adj’ed w/o**	**Perpl. with**	**Acc./adj’ed w/o**	**Acc. with**
N-Gram	37.3/35.7	31.8 (-8.9%)	.22/.25	.29 (+.04)
RNN	33.0/31.6	30.5 (-3.5%)	.26/.29	.31 (+.02)

Table 3: Perplexity and prediction accuracy without and with normalisation

Improvement is observed on both models, particularly in the simpler N-Gram model. This is presumably due to the room for compensating its weaknesses is greater, but it is encouraging that we see a gain in the RNN model too.

7.3 Transliteration

For transliteation the architecture is, as discussed in 6.2, such that the clustermodel kicks in as a plug-in predictor after segmentation by a language model, in our case an N-gram model. We evaluate its performance in comparison with two language models by letting them do the prediction too, 'base' model that is N-gram model, Okuno's (2016) RNN model. In Table 4, we show the accuracy figures of the baseline (N-Gram), RNN and our model (N-Gram + clustering module). The main results, 'homophone accuracy', are the ones confined to our set of homophones/heterographs, where a significant improvement is recorded for our model over the baseline, as well as RNN, model. Since there are other factors that affect input method qualities (particularly over different segmentations), it is not an overall win. The figures on the right show the overall accuracy, which includes the conversion accuracy of all words, not just target homophones, where we see roughly the same performance on RNN and the baseline + clustering. Nevertheless the results look promising, achieving a significant improvement if excluding these factors, which carry their overhead.

Model	**Acc, homophone**	Overall Acc
	(improvement against baseline)	
Baseline	.68	.39
RNN	.75 (+.07)	.44 (+.05)
Clustering	.84 (+.16)	.43 (+.04)

Table 4: Accuracy in homophone transliteration

Error examples indicative of each model's strengths/weakenesses include the following. On our bear/black-eye example with the contextually appropriate verbs, the baseline model made an error (on black-eye), while both RNN/clustering models got them right. The same seems to hold for infrequent cases as long as the context is in the vicinity, like ウィルスの抗体検査 ('virus antibody test'), for which only the baseline model outputs a wrong candidate, 交代 ('substitute'). The cluster model excels with the long-distance context, being the only model that gets our example in Section 4 on a Buddhist monk right. On the other hand, it is powerless in the transliteration possibilities spanning over different word tokens, such as マグロを包丁で解体 ('dismember with a knife a tuna' – lit. in word order), where the only winner is RNN, while the others return phonemically identical *phrase* 買いたい ('want to buy').

8 Concluding remarks

We described a method of word sense clustering which uses contextualised word embeddings, evaluate its meaning differentiation capacity and use the model for meaning-based normalisation of orthography and context-sensitive transliteration, with promising improvement.

We might emphasise that the proposed clustering procedure being unsupervised, it can be applied to any language. Though we focused on Japanese, where the homophone/homonym issue is most serious, the same procedure can be used in other languages for various purposes. The immediate application can be made to Chinese, which have similar challenges concerning orthography. We can apply normalisation perfectly well also to non-ideographic languages, with the proviso of making reference data available for evaluation.

The most serious outstanding problem concerns the issue of halting criterion for the optimal number of clusters. The clustering result is adequate enough *except* in homogeneity, which indicates oversplitting. We tried to find the optimum experimentally with various criteria but a better alternative is to use *Dirichlet Process* (Ferguson, 1973), which can find it in a more principled manner. Unfortunately, its computational cost has prevented us from finishing the experiment and constitutes a major future task.

References

Eneko Agirre and Aitor Soroa. 2007. SemEval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12, Prague, Czech Republic, June. Association for Computational Linguistics.

Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction. *CoRR*, abs/1905.12598.

Anne Cocos and Chris Callison-Burch. 2016. Clustering paraphrases by word sense. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1463–1472, San Diego, California, June. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Thomas Ferguson. 1973. Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230.

Kartik Goyal and Eduard Hovy. 2014. Unsupervised word sense induction using distributional statistics. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1302–1310, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Bo Han, Paul Cook, and Timothy Baldwin. 2011. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 49th Annual Meeting of ACL*.

Yafang Huang and Hai Zhao. 2018. Chinese Pinyin Aided IME, Input What You Have Not Keystroked Yet. *CoRR*, abs/1809.00329.

Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: Neural-based Chinese pinyin aided input method with customizable association. In *Proceedings of ACL 2018, System Demonstrations*, pages 140–145, Melbourne, Australia, July. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China, November. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2(1):198–218.

Yohei Kikuta. 2019. BERT pretrained model trained on japanese wikipedia articles. `https://github.com/yoheikikuta/bert-japanese`.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Field to Japanese morphological analysis. In *Proceedings of the conference on Empirical Method in Natural Language Processing*.

Taku Kudo, Hiroyuki Komatsu, Toshiyuki Hanaoka, Jun Mukai, and Yusuke Tabata. 2011. 統計的かな漢字変換システム Mozc (Mozc, a statistical kana kanji conversion system). In 言語処理学会第 *17* 回年次大会発表論文集 *(Proceedings of the annal meeting of the Japanese Association of Natural Language Processing)*. Japanese Association of Natural Language Processing.

Andrey Kutuzov and Elizaveta Kuzmenko. 2016. Neural embedding language models in semantic clustering of web search results. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3044–3048, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Mainichi Shimbun. 1995. Mainichi Newspaper CD-ROM 1995.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. SemEval-2010 task 14: Word sense induction &disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68, Uppsala, Sweden, July. Association for Computational Linguistics.

Diana McCarthy, Mariana Apidianaki, and Katrin Erk. 2016. Word sense clustering and clusterability. *Computational Linguistics*.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020. What do you mean, BERT? assessing BERT as a distributional semantics model. In *Proceedings of the Society for Computation in Linguistics*, volume 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Benjamin Muller, Benoit Sagot, and Djamé Seddah. 2019. Enhancing BERT for lexical normalization. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 297–306, Hong Kong, China, November. Association for Computational Linguistics.

National Institute of Japanese Language and Linguistics. 2003. Corpus of Spontaneous Japanese.

Yoh Okuno and Shinsuke Mori. 2012. An ensemble model of word-based and character-based models for Japanese and Chinese input method. In *Proceedings of the Second Workshop on Advances in Text Input Methods*, pages 15–28, Mumbai, India, December. The COLING 2012 Organizing Committee.

Yoh Okuno. 2016. Neural IME: Neural input method engine. `https://github.com/yohokuno/neural_ime`.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Simone Romano, Nguyen Xuan Vinh andJames Bailey, and Karin Verspoor. 2016. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June. Association for Computational Linguistics.

Itsumi Saito, Sadamitsu Kugatsu, Hisako Asano, and Yoshihiro Matsuo. 2014. Morphological analysis for Japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING 2014*.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2001. Estimating the number of clusters in a dataset via the gap statistic. *Journal of Royal Statistical Society*, 63:411–423.

Dmitry Ustalov, Denis Teslenko, Alexander Panchenko, Mikhail Chernoskutov, Chris Biemann, and Simone Paolo Ponzetto. 2018. An unsupervised word sense disambiguation system for under-resourced languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Nasser Zalmout, Kapil Thadani, and Aasish Pappu. 2019. Unsupervised neologism normalization using embedding space mapping. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 425–430, Hong Kong, China, November. Association for Computational Linguistics.