Why Not Simply Translate?
A First Swedish Evaluation Benchmark for Semantic Similarity

Tim Isbister
Peltarion
tim.isbister@peltarion.com

Magnus Sahlgren
RISE
magnus.sahlgren@ri.se

Abstract

This paper presents the first Swedish evaluation benchmark for textual semantic similarity. The benchmark is compiled by simply running the English STS-B dataset through the Google machine translation API. This paper discusses potential problems with using such a simple approach to compile a Swedish evaluation benchmark, including translation errors, vocabulary variation, and productive compounding. Despite some obvious problems with the resulting dataset, we use the benchmark to compare the majority of the currently existing Swedish text representations, demonstrating that native models outperform multilingual ones, and that simple bag of words performs remarkably well.

1 Introduction

Semantic Textual Similarity (STS) is a foundational concept in Natural Language Processing (NLP), with application in a wide range of tasks including Text Categorisation, Text Clustering, Text Summarisation, Recommender Systems, Information Retrieval, Question Answering, and so on. These, and other tasks, benefit from the ability to quantify the *semantic* similarity between two texts, t_1 and t_2. This is done by representing each text by vectors $\vec{t_1}, \vec{t_2}$ such that the similarity $sim(\vec{t_1}, \vec{t_2})$ is high if the texts are semantically similar and low if they are not. The text representations are often produced by using some type of *distributional* model (Sahlgren, 2008; Gastaldi, 2020), be it word embeddings or contextualized language models.

A main challenge in research on STS is how to evaluate the quality of the text representations. The arguably most straightforward way to evaluate semantic text representations is to use manually annotated data where pairs of texts are assigned a similarity score. The objective of an STS model is then to produce similarity scores that correlate with the human judgements. This has proven to be a useful and productive approach to promote development of STS models, in particular for English, where there exist high-quality testdata. For other languages, such as Swedish, the situation is not as simple, and there is currently no publicly available evaluation data to facilitate the development of Swedish STS models. This is a major bottleneck at the moment for Swedish NLP, which needs to be resolved. This paper presents a first simple step towards Swedish STS data, pending a more measured and rigorous approach.[1]

2 Data and Method

The arguably cheapest and most efficient way to produce a benchmark for semantic similarity in Swedish is to use machine translation to translate English STS data. In this paper, we use the English STS-B corpus from the GLUE benchmark,[2] since it is one of the standard evaluation resources for semantic similarity. The English STS-B data consists of sentence pairs with human similarity ratings that range from 5.00 from most similar to 0.00 for most dissimilar. We translate the English data to Swedish using the Neural Machine Translation (NMT) model provided by Google.[3] Table 1 shows the vocabulary size, lexical richness (or type-token ratio), as well as average word and sentence length of the original English and translated Swedish data. Note the increase in word length in the Swedish data, which is caused by compounds. Note also the increase in vocabulary size and lexical richness, which is likely due to artefacts from the machine translation (more about this in the

[1] https://www.vinnova.se/p/superlim-en-svensk-testmangd-for-sprakmodeller/
[2] http://ixa2.si.ehu.es/stswiki
[3] https://cloud.google.com/translate/docs/advanced/translating-text-v3

Language	Vocab size	Lexical richness	Avg.word length	Avg.sentence length
English	13 573	0.08	4.65	11.44
Swedish	19 229	0.12	5.21	10.73

Table 1: The English and Swedish data compared with respect to vocabulary, word length and sentence length.

next Section). The Swedish dataset is publicly available and can be accessed from Github.[4]

3 Error Analysis

There are of course a number of issues resulting from the machine translation process. One is the presence of anglicisms, where the translation is not literally incorrect, but where there exists a more conventional Swedish form. One example is the sentence "a plane is taking off," which is translated to "ett plan tar fart." Although it would be possible to use this construction in Swedish (the literal meaning is "a plane takes speed"), a more conventional translation would be "ett plan lyfter." The corresponding sentence pair in the STS-B data is "a plane is taking off" / "an air plane is taking off," which in the machine translated result becomes "ett plan tar fart" / "ett luftplan tar fart." Note the unconventional translation of "taking off" ("tar fart" instead of the more conventional "lyfter"), as well as the unconventional (but not strictly incorrect) term "luftplan" instead of the more conventional "flygplan." Even though this sentence pair may be regarded as pragmatically incorrect from a translation perspective, it is not obvious that this sentence pair would *not* work as an evaluation item for semantic similarity measures; the only difference between these two sentences is the compound "luftplan," which although being an unconventional (and somewhat archaic) term is not unrelated to the shorthand "plan." From this perspective, a maximum similarity score (in the case of STS-B, this means a score of 5.00) seems reasonable for the Swedish translation.

This type of vocabulary discrepancy might not affect the usefulness of the data, since the vocabulary typically remains in the same domain. Another example of a translation error that does not affect the usefulness of the data is the apparent inability of the Google machine translation API to correctly translate different verb tenses. One particularly problematic case seems to be the difference between simple present tense and present progressive, as in "peels" versus "is peeling," or "brushes" versus "is brushing." Such tense differences are normally not preserved in the Swedish data, where only the simple present tense is retained; i.e. both "peels" and "is peeling" are translated to "skalar" and not to "håller på att skala," which would be the correct progressive form. This can be regarded as a translation error, but it has no effect on the result, since these sentences always have a maximum similarity score in the STS-B data. The same consideration applies to other types of translation errors, where the resulting translation is nonsensical (or at least very contrived), such as the sentence "a person is folding a piece of paper," which becomes "en person fälls ett papper" (literally "a person is felled a paper"), but where the incorrect translation occurs in both translated sentences. Thus, as long as the translation errors are consistent, they have a limited effect on the usefulness of the data.

The majority of the inconsistencies in the machine translated material concerns vocabulary. In order to arrive at a quantitative measure of the vocabulary issues in the translated data, we compare its vocabulary to the biggest Swedish vocabulary we could find, which is the Swedish Skipgram model trained on the Swedish CoNLL17 corpus, available at the NLPL word embedding repository.[5] This vocabulary contains 3 010 472 words, a substantial part of which are preprocessing errors and other noise (due to the data being collected from the Internet). 82.77% of our test vocabulary can be found in the model. The other 17.22% contain both nonsensical translation errors ("afaict", "airstrike-ärendet", "arrestationen") as well as correct, but probably not very common, terms ("2006-versionen", "'aktiekursdetaljer", "antimissilförsvar"). Most of the 3 762 terms that do not occur in the NLPL vocabulary are compounds, which is perhaps not very surprising; a well-known challenge when counting vocabulary in compounding languages is that the number of possible compounds is very large, if not infinite.

[4] https://github.com/timpal01/sts-benchmark-swedish

[5] http://vectors.nlpl.eu/repository/20/69.zip

This poses a significant challenge for token-based models such as word embeddings, which are dependent on a comprehensive vocabulary. Models that have the capacity to include subword units and character n-grams, such as FastText and models based on wordpiece/BPE encoding are much better suited to handle this challenge. We therefore hypothesise that the machine translated data will work better for comparison of subword/character-based models than for token-based ones.

4 Experiments

Representation learning has been an enormously productive research area in recent years, with a progression from token-based embeddings to contextualized language models, which by now completely dominate representation learning for NLP. As a first application of the Swedish STS dataset, we compare a majority of the currently existing representation models for Swedish. This includes the following models:

TF: The arguably simplest form of Bag-of-Words (BoW) representation based on term frequency. We collect term frequencies from the training and development data, and simply apply the frequencies to the test data.

TF-IDF: BoW representation that weights term importance by the inverse document frequency. As with the TF representation, we count TF-IDF weights from the training and development data, and apply the weights to the test data. We use two versions in the supervised setting: one where we simply apply the IDF weights to the test data using words as tokens, and another one where we feature engineer the IDF representation to contain character n-grams ranging from 1 to 5 characters. To get a fixed size vector, the element-wise difference between the n-gram vectors are used to train a supervised Support Vector Regressor.

Word2Vec: Shallow token-based language model (Mikolov et al., 2013). We use the Skipgram model from the NLPL repository, which we have already introduced in Section 3. The vectors for sentences are obtained by averaging the embedding vector for each word.

fastText: A variant of Word2Vec that considers character n-grams of the context words (Grave et al., 2018). We use the CBOW model that has been trained on Common Crawl and Wikipedia.[6] As with Word2Vec, the vectors for sentences are obtained by averaging the embedding vector for each word.

BERT: Deep Transformer network trained using a masked language modeling objective (BERT stands for Bidirectional Encoding Representations from Transformers (Devlin et al., 2019)). We include both currently existing Swedish versions of BERT; KB/BERT from the Royal Swedish Library (Malmsten et al., 2020), and AF/BERT from the Swedish Public Employment Service. As sentence representation, we use the mean token representations from the last layer.[7]

SBERT: Uses a siamese setting where two BERT (or other types of Transformer) models are trained using Natural Language Inference (NLI) data in such a way that the training objective enforces similar representations for sentences with an entailment relation in the training data (Reimers and Gurevych, 2020). The resulting model has been demonstrated to produce useful sentence representations (hence the name Sentence-BERT, or SBERT in short) that outperform the standard BERT representations.

XLM-R: The RoBERTa Transformer model trained using a multilingual masked language modeling objective on massively multilingual data (Conneau et al., 2019).

LASER: Contextualized language model based on a BiLSTM encoder trained using a translation objective on parallel data (LASER stand for Language-Agnostic SEntence Representations (Schwenk et al., 2017)). The term "agnostic" is used because the model is claimed to handle more than 90 different languages. Since its not possible to retrain the whole architecture of the current LASER implementation, only the final representation can be used. We use this in a supervised manner by taking the element-wise difference from the embeddings and train a fully connected layer with Adam as optimizer.

LaBSE: A BERT variant trained on massively multilingual data using both masked language modeling and translation language modeling objectives. The resulting model is called Language-agnostic BERT Sentence Embedding (Feng et al., 2020). Similarly to LASER, the term "agnostic" is used because the model is claimed to handle more than 100 languages.

[6] https://fasttext.cc/docs/en/crawl-vectors.html

[7] Using the CLS representation produced consistently lower results in our tests.

Supervision	Model	Language	Test (sv)
	XLM-R	Multi	0.166
	Word2Vec	sv	0.374
	LaBSE	"Agnostic"	0.411
	KB/BERT	sv	0.419
	fastText	sv	0.420
	AF/BERT	sv	0.484
	LASER	"Agnostic"	0.704
NLI (en)	XLM-R ← SBERT	Multi	0.697
NLI (en) + STS (en)	XLM-R ← SBERT	Multi	0.801
NLI (en) + STS (en) + STS (sv)	XLM-R ← SBERT	Multi	0.808
STS (sv)	TF	sv	0.406
	TF-IDF	sv	0.547
	SVR-TF-IDF	sv	0.704
	AF/BERT	sv	0.714
	FFNN-LASER	"Agnostic"	0.764
	KB/BERT	sv	**0.825**

Table 2: Results for the various representations on the datasets used in these experiments

For each unsupervised model, we produce a fixed-sized vector for each sentence, and compare sentence pairs using cosine similarity, and for the supervised models the regression output is used. We use Pearson correlation coefficient to compare the resulting similarity measures with the gold labels of STS-B.

5 Results

Table 2 summarizes the results. Note that there is no consistent difference between token-based embeddings and contextualized ones, when there is no supervision for the sentence representations. In particular XLM-R underperforms in the unsupervised case, and the most recent LaBSE model does no better than fastText embeddings. The best model in the unsupervised setting is LASER, which seems to produce useful sentence representations for Swedish even without supervision.

Using SBERT significantly improves the performance of XLM-R, which is expected. Adding finetuning with the English STS-B data further improves the performance, and adding Swedish finetuning on top improves the result even further. This demonstrates the capacity for cross-lingual transfer using multilingual models. Adding supervision to the native Swedish models improves their performance, and our best score is reached by the KB/BERT model finetuned on the Swedish data. Note that the simple BoW model with Support Vector Regression reaches a performance of 0.704, which is remarkably competitive considering the enormous difference in computational cost between this and the other models.

6 Conclusions

Machine translation introduces a number of issues into the data, mostly concerning vocabulary. We argue that this is problematic for token-based models, but should be manageable for subword- and character-based models. We thus do not recommend that the machine translated STS-B data is used with standard word embeddings, but language models that rely on wordpiece/BPE tokenisation should be able to handle the vocabulary issues, and as such should be amenable to comparison using the Swedish STS-B dataset introduced in this paper.

Due to the high prevalence of translation errors, we do not recommend that the translated data is used to train or finetune models for downstream deployment. The translation errors likely have a limited effect for comparison between different models, but it is unclear what effects they might have for downstream application.

Acknowledgements

This work was partly supported by Vinnova under grant 2019-02996. We wish to thank anonymous reviewer #2 for valuable comments.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.

Juan Luis Gastaldi. 2020. Why can computers understand natural language? the structuralist image of language behind word embeddings. *Philosophy & Technology*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden – making a swedish bert.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54.

Holger Schwenk, Ke M. Tran, Orhan Firat, and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. *CoRR*, abs/1704.04154.