Exploring word embeddings and phonological similarity for the unsupervised correction of language learner errors

Ildikó Pilán
Språkbanken, University of Gothenburg
Sweden
ildiko.pilan@gu.se

Elena Volodina
Språkbanken, University of Gothenburg
Sweden
elena.volodina@gu.se

Abstract

The presence of misspellings and other errors or non-standard word forms poses a considerable challenge for NLP systems. Although several supervised approaches have been proposed previously to normalize these, annotated training data is scarce for many languages. We investigate, therefore, an unsupervised method where correction candidates for Swedish language learners' errors are retrieved from word embeddings. Furthermore, we compare the usefulness of combining cosine similarity with orthographic and phonological similarity based on a neural grapheme-to-phoneme conversion system we train for this purpose. Although combinations of similarity measures have been explored for finding correction candidates, it remains unclear how these measures relate to each other and how much they contribute individually to identifying the correct alternative. We experiment with different combinations of these and find that integrating phonological information is especially useful when the majority of learner errors are related to misspellings, but less so when errors are of a variety of types including, e.g. grammatical errors.

1 Introduction

During the language acquisition process, learners often use word forms which deviate, in one way or another, from a standard that native (L1) speakers usually adhere to. These deviations, also referred to as *errors* or *non-normative* forms, result often in differences in spelling which can lead to forms that might not exist in a language (*non-word* errors). Inferring the intended meaning and the correct word form for such errors can be challenging both for humans and for machines.

In previous work, a number of approaches have been tested for the automatic correction of language learner errors, which rely often on annotated data (Ng et al., 2014; Mishra and Kaur, 2013). For normalizing native speakers' non-standard use of spelling, however, a recent direction being explored is the use of word embeddings as an unsupervised solution (Bertaglia and Nunes, 2016; Fivez et al., 2017). A major advantage of this approach is that it does not require annotated training data. Such methods rely on the intuition that semantically similar words are grouped close to each other in the vector space. Incorporating character n-grams when building word representations completes this type of similarity with orthographic and morphological relatedness (Bojanowski et al., 2016). This can be useful for detecting correction candidates for those types of errors that involve only a slight variation of word forms, such as spelling and inflectional errors. Cosine similarity is often combined with other lexical similarity measures (Bertaglia and Nunes, 2016), their individual contribution and interaction, however, remains less explored. The research questions that we address in this article are:

- RQ1: How useful are word embeddings based on character n-grams for retrieving correction candidates for language learners' errors?
- RQ2: Does capturing phonological similarities between sounds provide helpful additional information for identifying corrections?

Proceedings of Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, pages 119–128
Santa Fe, New Mexico, USA, August 25, 2018.

- RQ3: What combination of different similarity measures is most efficient in this context?

The usefulness of word embeddings with character n-grams has not been previously explored specifically for correcting errors made by second or foreign language (L2) learners. While there might be similarities between the type of spelling errors that L1 and L2 speakers make (e.g. based on the position of keys while typing), in the case of the latter category grammatical errors may also occur. Moreover, L2 spelling errors may be often induced also by sound similarity or orthographic differences between L1 and L2.

Since data annotated with information about L2 errors and their correction is scarce for most languages, exploring unsupervised methods in this context is particularly valuable. This is the case also for Swedish, the target language of our experiments. As an unsupervised solution, we evaluate embeddings where words are represented as the sum of their character n-grams for L2 error correction by inspecting how often correction candidates occur among the most similar words based on cosine similarity. We compare embeddings created using a large collection of Wikipedia articles with embeddings trained on a smaller amount of *specialized* corpora related to L2 learning combined with blog texts. We find that the latter alternative provides twice more often the correction among the set of most similar words than the former.

One of the obstacles of spelling words correctly is that different graphemes can be used to encode the same sound in a certain language. This similarly in pronunciation, however goes beyond the type of information that a similarity measure based on character n-grams and orthography could provide. Therefore, we propose a phonological similarity measure for capturing this aspect. To address RQ2, we train a grapheme-to-phoneme conversion system based on neural networks to be able to map even out-of-vocabulary (OOV) words, such as non-word errors, to a phonological representation. We show that this system improves on the results of a previous rule-based attempt at solving this task for Swedish. We then compute the similarity based on a Levenshtein distance discounted for phonological relatedness between the phonological representation of each non-word error and that of the intended word. Furthermore, we investigate the correlation between different similarity measures and find that phonological similarity correlates less with cosine and orthographic similarity than these two with each other.

In relation to RQ3, we compare cosine, orthographic and phonological similarity and investigate their interaction for finding the intended word for an L2 error. We aggregate cosine similarity and the other two similarity measures individually and in combination and observe how the ranking of the correct word changes compared to other most similar words retrieved from word embeddings. Our results indicate that combining cosine and orthographic similarity worked best for the variety of errors found in learner essays, whist for errors collected from spelling exercises summing all three measures, or combining cosine and phonological similarity was more efficient.

This work has also a number of engineering contributions, which are being made freely available.[1] This includes (i) a word embedding incorporating character n-grams, trained on a combination of L2 relevant corpora and blog texts; (ii) a grapheme-to-phoneme conversion system for Swedish based on a large lexical database and deep learning methods; (iii) the implementation of a phonological similarity measure based on binary phonological features rooted in linguistic theory.

This article is structured as follows. In section 2, we present previous work related to error correction, which is followed by the description of a small evaluation dataset (Section 3) used to measure the usefulness of the proposed techniques. We then describe the word embeddings (Section 4) and the grapheme-to-phoneme conversion system (Section 5) created, which are at the basis of the similarity measures used for finding optimal correction candidates. Section 6 details the results of our experiments on the usefulness of the similarity measures. Finally, we conclude our paper in Section 7.

2 Related work

In this section, we first summarize previous literature on error correction with the use of NLP in the educational domain. Then, we focus specifically on the use of word embeddings for spelling normalization and previous attempts at solving this task with different methods for Swedish.

[1] The resources are available at `https://github.com/IldikoPilan/swell-norm`

2.1 Error correction in educational NLP applications

Throughout the language learning process, learners produce a range of written responses which vary in size and quality depending on the specific task and learners' proficiency level. A common scale of proficiency levels is the CEFR, the Common European Framework of Reference for Languages (Council of Europe, 2001). The CEFR proposes a six-point scale of proficiency levels which ranges from A1 (beginner) to C2 (advanced) level.

Learner-written texts are challenging to process automatically since, unlike the standard language texts used for training most NLP tools, they often contain errors. This is especially problematic for texts written by lower proficiency learners where the amount of such errors can have a substantial impact on the accuracy of automatic analyses. Both rule-based and statistical methods have been explored for the automatic detection and correction of errors, including finite state transducers (Antonsen, 2012) and different hybrid systems proposed in connection with the CoNLL Shared Task on grammatical error correction for L2 English (Ng et al., 2014). Most of the competing systems of this Shared Task re-used existing spell checking systems in their L2 error correction pipeline. Solutions for dealing specifically with L2 misspelling remained, however, less explored.

Language learning errors provide useful insights into L2 learners' development and have therefore been used for assessing writing quality. Yannakoudakis et al. (2011) present experiments for automatically predicting overall, human-assigned scores for texts written by L2 English test takers at upper-intermediate level. Error-rate features showed a high correlation with these scores. E-rater (Burstein, 2003) is a commercial essay scoring system that measures writing quality based on a variety of linguistic features. Also in this system, grammatical accuracy is used as an indicator of quality alongside the topical relevance of the vocabulary used and features based on discourse analysis.

2.2 Spelling error correction with word embeddings

Word embeddings represent words in a vector space by grouping semantically similar words near each other based on the idea that words that share similar contexts are semantically related (Baroni et al., 2014). We can measure how closely related two or more words are based on cosine similarity from these representations. Word embeddings are useful for a number of lexical semantic tasks such as detecting synonyms and disambiguating word senses, see e.g. in Iacobacci et al. (2016). Recently, the usefulness of these models has been also explored for spelling error correction. Bertaglia and Nunes (2016) explore word embeddings for normalizing noisy user-generated content for Brazilian Portuguese. The authors collect correction candidates from the embeddings based on cosine similarity and rank them by computing a combined similarity score. This includes, besides cosine similarity, a lexical similarity measure consisting of edit distance, longest common sub-sequence and diacritical similarity. Fivez et al. (2017) present a similar method for spelling error correction in English clinical text based on character n-gram embeddings. Correction candidates are collected from a lexicon based on both graphical and phonological distance and the candidate maximizing cosine similarity with the context words appearing around the error is chosen as correction. The authors find that this approach outperformed existing off-the-shelf systems.

2.3 Spelling error correction for Swedish

Grigonyte and Hammarberg (2014) present a method for automatically identifying those misspellings made by L2 Swedish learners which are induced by a similarity in pronunciation. In the data[2] analyzed by the authors, 21% of L2 misspellings were related to pronunciation similarities. They propose a model based on a small dataset of errors and their corrections as well as a language model for distinguishing pronunciation-related misspellings and find that incorporating additional information in their model would be required for an improved performance.

Stymne et al. (2017) propose an annotation layer incorporating error correction for a corpus of L1 and L2 Swedish student writings. In their pilot experiment, correction candidates are collected from a smaller amount of annotated data and a lexical resource using a Levenshtein distance with discounted weights

[2] We investigated the availability of this data, but did not receive access to it up to the time of writing.

for frequently occurring errors. When several candidates are available, the most frequent correction candidate is chosen as correction. The authors report an accuracy of 71% without the use of annotated training data for the whole corpus and 73% for texts written by L2 learners with a model using training data.

3 Evaluation data for L2 Swedish error correction

To evaluate the usefulness of word embeddings and similarity measures, we combined L2 Swedish learner error data from two different sources: a corpus of learner essays and logged spelling exercises. From the SweLL learner corpus (Volodina et al., 2016), we collected errors from 24 randomly sampled essays between CEFR levels A1 – B1 from the SpIn sub-corpus. Learners' native languages included Romanian, Vietnamese, Somali, Tigrinya, Dari, Latvian, Thai, Mandarin Chinese, Kurdish, Swahili, Albanian and Arabic. During the manual error correction, each non-lemmatized token was analyzed and, if they were errors, they were manually corrected. Only errors for which the intended token could be unambiguously determined were included. Errors with capitalization, foreign words and containing @, signaling unintelligible handwritten characters, were excluded.

Besides essays, we collected errors also from spelling exercise logs (SpellEx) from a Swedish language learning platform, *Lärka*[3]. We collected non-word errors from the responses of the 10 language learners (ranging from beginner to advanced level) who participated in a previous evaluation of the platform (Pijetlovic, 2013; Volodina and Pijetlovic, 2015). Word segmentation errors and other errors consisting of valid inflectional forms were not considered. The size of the dataset in terms of number of L2 errors and their distribution across data sources and CEFR levels is presented in Table 1. Digits in parenthesis for SpIn indicate the number of individual essays.

	A1	**A2**	**B1**	**Sub-total**	**Total**
SpIn	82 (8)	62 (9)	58 (7)	202	**455**
SpellEx	NA	NA	NA	253	

Table 1: The number of corrected non-word L2 errors in the dataset.

It is worth noting that, while the spelling exercise log part of the data consisted mainly of L2 spelling errors, the errors collected from the essays displayed a wider variety of error types including grammatical and vocabulary errors. Duplicate pairs of error and intended word have been removed from the dataset, thus the numbers in Table 1 refer to unique error types.

4 Word embeddings enhanced with character n-grams for learner error correction

In this work, we use FastText (Bojanowski et al., 2016) for training embeddings. FastText is a recently proposed approach that enhances traditional word-based vectors by representing each word as a bag of character n-grams. Incorporating this type of subword information, besides semantic relatedness, allows for capturing also orthographic and morphological similarity.

We compare pre-trained word vectors[4] using a large amount of Swedish Wikipedia articles with word vectors we trained specifically for the purpose of finding L2 error correction candidates, which can be an alternative (or a complement) to lexicon-based lookups. We base our embeddings on corpora which are more similar to what L2 learners produce in terms of topic or complexity. These include a small set of specialized corpora (*SpecC*) combining easy-to-read texts (Heimann Mühlenbock, 2013), L2 coursebook texts (Volodina et al., 2014) and L2 learner essays (Volodina et al., 2016) We combine these with a large amount of blog texts to ensure a sufficiently large lexical basis for our representations. The blog texts were collected via the corpus query system, *Korp*[5] (Borin et al., 2012). Compared to Wikipedia, the topic of these blog texts is more similar to that of learner texts dealing often with everyday topics, which is part

[3] `https://spraakbanken.gu.se/larka/`

[4] `https://fasttext.cc/docs/en/pretrained-vectors.html`

[5] `https://spraakbanken.gu.se/korp/`

of language learning curricula according to CEFR, especially between beginner (A1) and intermediate levels (B1) (Council of Europe, 2001). Moreover, blog texts might contain misspellings, which, although not necessarily produced by L2 speakers, can potentially increase the usefulness of this type of data for detecting correction candidates for L2 errors.

The total number of tokens in the combination of corpora used for our embeddings was 25 million tokens (8% being from SpecC), which resulted in a vocabulary size of 307,349 word forms. This is a considerably smaller vocabulary size than that of the pre-trained Wikipedia embeddings consisting of 1,143,273 word forms. The pre-trained embeddings were based on the skip-gram model and they used the default parameters and 300 dimensions. We train embeddings using Continuous Bag-of-Words model (CBOW) which can be used also to predict target words from the context (Mikolov et al., 2013). We preserve most parameters with their default values, except for four exceptions. We set the number of dimensions to 300 and we considered n-grams between 2 and 5 (instead of the default 3 and 6), in attempt to better handle short words, which are common at lower proficiency levels. We also reduce the minimum number of occurrences for a word to be included to 2 and add a duplicate copy of the small amount of learner-written texts in the training data.

5 Phonological similarity based on Levenshtein distance

Levenshtein (edit) distance is a common measure of the difference between strings in terms of the minimum number of single character edits required to map one string into another (Levenshtein, 1966). These edits can consist of deletions, additions and substitutions. The traditional version of this measure makes a binary decision about whether a pair of characters match. Some characters, however, are more likely to be misspelled as certain characters than others. This can be either due to the frequent omission of highly language-dependent diacritics or due to a similarity in the sound encoded by different graphemes. An example for the former case is: *traffas* instead of *träffas* 'meet'. To account for phonological differences, we first train a grapheme-to-phoneme conversion system that can map a string (even if it contains errors) to its phonological representation, which we describe in the next sub-section.

5.1 A neural grapheme-to-phone conversion for Swedish

Grapheme-to-phoneme (g2p) conversion is a task consisting of transforming the orthographical representation of words into their phonological equivalent. This is an important building block of, among others, text-to-speech (TTS) applications. A number of approaches have been explored in the past to tackle this conversion, which include both rule-based and data-driven methods. Torstensson (2002) presents a rule-based approach to g2p conversion for a Swedish TTS system and reports a rate of correctly transcribed words of 78% when testing on about 3700 unique words.

In this work, we train a g2p conversion system for Swedish based on a g2p toolkit[6] available within CMUSphinx (Yao and Zweig, 2015), an open source software mainly aiming at speech recognition. This g2p toolkit uses TensorFlow (Abadi et al., 2016) and a transformer model[7] relying on an attention mechanism. The toolkit has been successfully applied by a number of large companies (Yao and Zweig, 2015). The advantage of this g2p system is that, as opposed to other approaches, it does not require phoneme to grapheme alignments. Instead, mappings are learned directly from a list of words consisting of pairs of orthographic and phonemic representations for each word.

We use the Swedish lexical database of the Nordisk språkteknologi holding AS (NST)[8] to collect orthographic forms and their transcriptions encoded in Speech Assessment Methods Phonetic Alphabet (SAMPA). Before feeding the transcriptions to the g2p system, we automatically segment them as the example CMU dictionary for English. Length information is retained in the transcriptions, but we remove other non-phone related information (e.g. syllable and compound constituent boundaries). When collecting information from the dictionary, we select only base forms that were assigned a Swedish language code[9] and that were not tagged as garbage entries or as acronyms. All orthographic forms are

[6] `https://github.com/cmusphinx/g2p-seq2seq`

[7] `https://ai.googleblog.com/2017/06/accelerating-deep-learning-research.html`

[8] `https://www.nb.no/sprakbanken/show?serial=sbr-22`

[9] The NST dictionary contains some non-Swedish entries from other languages.

normalized to lower case and duplicates are removed, which results in a dataset consisting of 103,026 pairs of orthographic forms and their transcriptions.

We train a model with the default parameters using 90% of the extracted NST data and comparing two different sizes for the hidden units (64 and 512). We then measure the performance of the system on the held-out 10% (10,302) of the data in terms of accuracy and word error rate (WER). In Table 2, we compare our results with those presented for English using the same implementation. As a baseline (BL) for Swedish, we use the results reported in Torstensson (2002).

	Swedish-BL	**Swedish**		**English**	
# units	NA	64	512	64	512
Accuracy (%)	78.3	82.6	**86.6**	68.7	76.7
WER (%)	21.7	17.4	**13.4**	31.3	23.3

Table 2: The performance of the Swedish g2p conversion system.

Our system outperforms considerably the previously reported results for Swedish for this task. Moreover, our neural models for Swedish achieve a higher accuracy than those for English with both 64 and 512 units. Besides some differences in the underlying data, this may be due to the fact that Swedish orthography is closer to the pronounced forms of words.

5.2 Measuring distance between sounds based on binary phonological features

To be able to measure distance between sounds, we adopt the binary feature representation described in Hayes (2009), where each phoneme is characterized across 26 dimensions divided into three categories: manner, place and laryngeal features. For a detailed description of each feature and their value per phoneme, see Hayes (2009).

Some peculiar sounds in Swedish that occur less commonly in other languages include retroflex sounds (e.g. ʈ, ɖ) and the doubly articulated postalveolar-velar fricative (/ɧ/). Since the latter was missing from the phonemes listed in the feature charts from Hayes (2009), we deduced its values by combining the features of the two sounds it is composed of according to IPA,[10] namely /ʃ/ and /x/.

5.3 From Levenshtein distance to a phonological similarity measure

We employ a memory efficient version of the Levenshtein distance (LD) using two row matrices.[11] We replace the original *cost* of 1 for non-matching characters with a value that expresses a phonological similarity between two phonemes. The similarity is computed based on the phonological features of these sounds. The cost for this phonological distance (PH-LD) is computed as the ratio of features with matching values divided by the total number of features relevant for at least of the two sounds being compared. In section 6, we report results for both the traditional version of LD and the proposed phonological LD.

We transform both types of LD distances into a normalized similarity measure to make it more comparable with cosine similarity values. We compute orthographic similarity as $1 - \frac{LD}{N_{char}}$ and phonological similarity as $1 - \frac{PH-LD}{N_{phon}}$, where N_{char} and N_{phon} stand for the number of characters and phonemes in the word respectively.

6 Investigating the usefulness of similarity measures for L2 error correction

6.1 The efficiency of retrieving L2 error corrections based on cosine similarity

To start with, we evaluate the usefulness of word embeddings for retrieving the intended word for learner errors. Table 3 shows the percentage of errors for which the correction appeared among a varying number

[10] https://www.internationalphoneticassociation.org/content/full-ipa-chart

[11] We used Christopher P. Matthews' Python implementation available at https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance#Python

of *k* most similar words.

	SpecC+Blogs			Wikipedia		
Top *k*	**SpIn**	**SpellEx**	**Avg**	**SpIn**	**SpellEx**	**Avg**
10	40.6	50.2	45.4	18.8	24.5	21.65
25	49.0	57.3	53.15	24.8	28.1	26.45
50	59.4	63.2	61.3	28.7	31.6	30.15
100	64.4	66.4	**65.4**	31.2	34.4	**32.8**

Table 3: Percentage of corrections occurring among the *k* most similar words.

The word embeddings created specifically for L2 error correction purposes (SpecC+Blogs) based on data relevant for the task proved to be considerably more useful for retrieving the intended word for learner errors than the pre-trained Wikipedia embeddings. This task-specific embedding contained approximately twice more often (or more) the intended word than its counterpart at all levels of *k*. Furthermore, word embeddings were more suitable for finding the correct alternative for the errors collected from the spelling exercises even though words from the essay corpus were used as part of the training data for the embeddings. This may be due to the fact that the errors in the essays include a higher percentage of grammatical errors besides spelling errors and embeddings enhanced with character n-grams may be more useful for the latter type.

6.2 Relationship between similarity measures

In the next step, we compare the average value of each similarity measure in the two different sources of L2 error data to investigate whether there is a difference in their usefulness.

	CosSim	**OrthSim**	**PhonSim**
SpIn	0.683 (0.181)	0.773 (0.115)	0.904 (0.100)
SpellEx	0.747 (0.184)	0.773 (0.142)	0.918 (0.107)

Table 4: Mean and standard deviation values (in parenthesis) for different similarity measures.

As Table 4 shows, the mean values of all three similarity measures are higher for the SpellEx data, containing mostly spelling errors. The similarity measures we investigate here seem therefore more efficient for capturing the similarity with an intended word for spelling errors rather than for a variety of different error types. Normalized phonological similarity values are, on average, higher than the values of the other two similarity measures, but it is hard to draw conclusions from comparing directly the average values per measure since they are computed differently.

To understand how much additional information the different similarity measures carry compared to each other, we measure also the Spearman's correlation (ρ) between them pairwise. We find that all three similarity measures positively correlate[12] with each other, but to a varying degree. Cosine similarity correlates more with orthographic ($\rho = 0.693$) than with phonological similarity ($\rho = 0.449$). Word embeddings do indeed incorporate character n-grams which rely on the same type of information as the one at the basis of orthographic similarity. Phonological similarity correlates less both with cosine and with orthographic similarity ($\rho = 0.449$ and $\rho = 0.472$ respectively). Some types of errors produce, in fact, an orthographic dissimilarity, but not a phonological one. These include errors with double consonants and different graphemes mapping to the same phoneme. In Table 5, we provide some examples for pairs of words with orthographic and phonological forms to illustrate these cases. (Phonological LD values, omitted from the table are 0 in all cases.)

[12] For all correlations $\rho =< 0.001$ holds.

Error		Correction		Translation	LD	Type of error
Orth	Phon	Orth	Phon			
b**e**ttre	b E t r e	_b**ä**ttre_	b E t r e	‘better’	1	wrong grapheme
**tj**eck	s’ E k	_**ch**eck_	s’ E k	‘check’	2	wrong grapheme
fortsä**t**a	f U t‘ s‘ E: t a	fortsä**tt**a	f U t‘ s‘ E: t a	‘continue’	1	consonant doubling

Table 5: Examples of errors producing orthographic, but not phonological distance.

As we showed in this section, phonological similarity has the potential to provide useful additional information when compared to the other two similarity measures. In the next sub-section, we investigate whether combining orthographic and phonological similarity improves the cosine similarity-based rank of an intended word for L2 errors.

6.3 The efficiency of the combination of different similarity measures

When using similarity measures for error correction purposes, a commonly adopted option is to choose the correction candidate that maximizes the different types of similarities, see e.g. Bertaglia and Nunes (2016). In this section, we investigate whether adding orthographic and phonological similarity can improve the initial, cosine similarity-based rank of an L2 error correction. We consider only those errors here for which the correction appeared among the 50 most similar words in the word embeddings trained on the combination of specialized corpora and blog texts. These errors were 280 in total from the two data sources. Table 6 presents the average ranks of the corrections and their standard deviation based on cosine similarity and their combination with the other measures. (A lower rank indicates a higher degree of similarity.)

	CosSim	CosSim+OrthSim	CosSim+PhonSim	CosSim+OrthSim+PhonSim
SpIn	6.91 (10.1)	**1.95 (1.71)**	2.32 (3.23)	2.61 (3.83)
SpellEx	10.83 (12.99)	2.82 (3.40)	3.54 (5.15)	**1.62 (1.54)**

Table 6: Average ranks per similarity measures.

On average, the best ranking was obtained by the combination of cosine and orthographic similarity for the essay errors from SpIn. For the spelling errors, however, incorporating also phonological information achieved the best results in terms of average ranks.

In the last set of experiments, we explore how often combining cosine similarity with orthographic and phonological similarity improved, worsened or had no impact on the cosine similarity-based ranking of the corrections. Table 7 presents the percentage of errors for each of these three types of effects per similarity measure combination. We find that combining cosine similarity with orthographic or phonological similarity boosts the ranking of the corrections in more that half of the cases. The results from Table 7 confirm the ones from Table 6: adding phonological similarity information boosts the ranks of corrections only for the spelling error data, but not for the mixed type of L2 errors from SpIn. Moreover, relying on more than one similarity measure seems to be more beneficial for the wider spectrum of errors occurring in learner essays (SpIn) where measure combinations yielded an improvement in ranking on average for ca. 10% more of the errors than for SpellEx.

7 Conclusion

In this paper, we explored different similarity measures and their interaction for the purposes of correcting errors made by language learners. We presented word vectors created for this purpose using task-relevant corpora which proved to be more efficient for retrieving corrections than pre-trained embeddings based on Wikipedia. Since almost 4 out of 10 times the correction did not appear among

		Improvement	No change	Drop
CosSim+OrthSim	SpIn	**64.86**	27.02	8.10
	SpellEx	48.99	44.96	6.04
CosSim+PhonSim	SpIn	61.26	22.52	16.21
	SpellEx	52.34	34.89	12.75
CosSim+OrthSim+PhonSim	SpIn	63.06	27.02	9.90
	SpellEx	**53.02**	39.59	7.38

Table 7: Effect of combining similarity measures.

the most similar words in the word embeddings, dictionary lookups for candidates within a certain edit distance would be a useful complement.

Furthermore, we trained a neural g2p system for computing phonological similarity between transcribed errors and their corrections which outperformed previously reported results targeting this task for Swedish. Although based on a similar type of information, complementing cosine similarity with orthographic similarity yielded considerable improvements for the ranking of corrections for a varied type of L2 errors found in essays. Adding phonological similarity information to cosine similarity, on the other hand, proved more useful for data containing spelling errors. The similarity measures described can be easily re-used for other languages with the availability of a lexicon with pairs of orthographic forms and their phonological transcription and by complementing the list of sounds and their phonological features with any potentially missing language-specific sound. The trained word embeddings, the Swedish g2p system and the implementation of the phonological similarity measure have been made publicly available to foster code reuse and replicability.

Future work could explore additional methods for both the retrieval of correction candidates for L2 errors and for ranking them. These could include, among others, language models, diacritical symmetry and a longest common subsequence measure. Moreover, the usefulness of the measures described could be investigated further with additional evaluation data both for Swedish and for other languages.

Acknowledgements

This work has been carried out within the "SweLL - research infrastructure for Swedish as a second language" project funded by the Swedish Foundation for Humanities and Social Sciences. We would also like to thank the Royal Swedish Academy of Letters, History and Antiquities for providing a travel grant to the first author via the Wallenberg Foundation.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Lene Antonsen. 2012. Improving feedback on L2 misspellings – an FST approach. In *Proceedings of the workshop on NLP for Computer-Assisted Language Learning*, pages 1–10. Linköping University Electronic Press.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247.

Thales Felipe Costa Bertaglia and Maria das Graças Volpe Nunes. 2016. Exploring word embeddings for unsupervised textual user-generated content normalization. In *Proceedings of the 2nd Workshop on Noisy User-generated Text*. COLING.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp – the corpus infrastructure of Språkbanken. In *LREC*, pages 474–478.

Jill Burstein. 2003. The e-rater Scoring Engine: Automated Essay Scoring with Natural Language Processing. *Lawrence Erlbaum Associates, Inc.*

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment.* Press Syndicate of the University of Cambridge.

Pieter Fivez, Simon Šuster, and Walter Daelemans. 2017. Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embedding. In *16th Workshop on Biomedical Natural Language Processing of the Association for Computational Linguistics*, pages 143–148.

Gintare Grigonyte and Björn Hammarberg. 2014. Pronunciation and spelling: the case of misspellings in Swedish l2 written essays. In *6th International Conference on Human Language Technologies-The Baltic Perspective (Baltic HLT), Kaunas, Lithuania, September 26-27, 2014*, pages 95–98. IOS Press.

Bruce Hayes. 2009. *Introductory Phonology.* John Wiley & Sons.

Katarina Heimann Mühlenbock. 2013. I see what you mean—assessing readability for specific target groups. *Data linguistica*, (24).

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 897–907.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

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

Ritika Mishra and Navjot Kaur. 2013. A survey of spelling error detection and correction techniques. *International Journal of Computer Trends and Technology*, 4(3):372–374.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant, editors. 2014. *Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task on grammatical error correction.* Association for Computational Linguistics, Baltimore, Maryland.

Dijana Pijetlovic. 2013. Swedish spelling game: Developing Swedish spelling exercises on the ICALL platform Lärka using Text-to-Speech technology. Master's thesis, University of Gothenburg.

Sara Stymne, Eva Pettersson, Beáta Megyesi, and Anne Palmér. 2017. Annotating errors in student texts: First experiences and experiments. In *Proceedings of the Joint 6th Workshop on NLP for Computer Assisted Language Learning and 2nd Workshop on NLP for Research on Language Acquisition at NoDaLiDa, Gothenburg, 22nd May 2017*, number 134, pages 47–60. Linköping University Electronic Press.

Niklas Torstensson. 2002. Grapheme-to-phoneme conversion, a knowledge-based approach. *Speech Music and Hearing TMH-QPSR-Fonetik*, 44:117–120.

Elena Volodina and Dijana Pijetlovic. 2015. Lark trills for language drills: Text-to-speech technology for language learners. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 107–117.

Elena Volodina, Ildikó Pilán, Stian Rødven Eide, and Hannes Heidarsson. 2014. You get what you annotate: a pedagogically annotated corpus of coursebooks for Swedish as a second language. In *Proceedings of the 3rd workshop on NLP for Computer Assisted Language Learning*, pages 128–144.

Elena Volodina, Ildikó Pilán, Ingegerd Enström, Lorena Llozhi, Peter Lundkvist, Gunlög Sundberg, and Monica Sandell. 2016. SweLL on the rise: Swedish learner language corpus for European Reference Level studies. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation.*

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1*, pages 180–189. Association for Computational Linguistics.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196.*