Armin Hoenen, Lela Samushia

Gepi: An Epigraphic Corpus for Old Georgian and a Tool Sketch for Aiding Reconstruction

In the current paper, an annotated corpus of Old Georgian inscriptions is introduced. The corpus contains 91 inscriptions which have been annotated in the standard epigraphic XML format EpiDoc, part of the TEI. Secondly, a prototype tool for helping epigraphic reconstruction is designed based on the inherent needs of epigraphy. The prototype backend uses word embeddings and frequencies generated from a corpus of Old Georgian to determine possible gap fillers. The method is applied to the gaps in the corpus and generates promising results. A sketch of a front end is being designed.

1 The Old Georgian Corpus

Basis for the corpus are the transcriptions present on the TITUS web thesaurus, Gippert (1995).[1] 91 inscriptions have been transcribed into digital form and annotated. The corpus comprises Old Georgian inscriptions with the oldest dated to the 5^{th} century A.D. written in Old Georgian Majuscle (Asomtavruli). However, some of the inscriptions stem from the new Georgian period and are written in the modern version of the alphabet (Mxedruli). The majority of inscriptions are building inscriptions (churches), yet there are some gravestone inscriptions and inscribed crosses and other objects. Of special importance for regional and national history are people mentioned mostly on gravestones and correlated data from the inscriptions. As Georgian has been written in three alphabets throughout its history, all inscriptions have been transcribed into the modern version of the alphabet in previous projects.

1.1 Corpus generation

Whilst the corpus is online and accessible via the Titus archive, a translation and annotations have been added. Additionally, the corpus has been transformed into the TEI-format in a way conforming to EpiDoc guidelines. EpiDoc, according to their website[2] is "an international, collaborative effort that provides guidelines and tools for encoding scholarly and educational

[1] http://titus.fkidg1.uni-frankfurt.de/texte/etcg/cauc/ageo/inscr/carcera/carce.htm
[2] https://sourceforge.net/p/epidoc/wiki/About/:last accessed on 07.02.2017

editions of ancient documents" which originated from an effort for publication of ancient inscriptions. In technical terms it uses a subset of the TEI. EpiDoc provides guidelines for the encoding of ancient documents, which the Old Georgian Corpus follows.

Each inscription is encoded in its own tei-xml file in order to ensure complete informativity on metadata and textual levels. The header contains meta information such as language, alphabet, place and time of the inscription as well as a link to its images if available on the TITUS web thesaurus which hosts the inscriptions electronically prepared at the National Museum of Georgia for the Georgian National Corpus (GNC), which they are part of.[3] The body of the document contains the four text divisions typical for EpiDoc: edition, translation, commentary and bibliography.

Annotations are applied to the text in the modern transcription. This transcription forming the TITUS base text previously already included expansions of abbreviations, fillers of gaps, most probable readings of unclear letters, letters the scribe had omitted and so forth (the canon of epigraphic annotation). The modern trancription thus displays one reconstructed text version for the inscription (where reconstruction was possible) and is consequently stored in the text division *edition*. Besides, each file provides the original characters (similar to the text in majuscules in Latin) preserving original linebreaks. Alongside, in a separate text division a full English translation is provided, which has been newly compiled and added to the corpus. People, titles, places and dates have been annotated in order to enable semantic technologies at later stages. Named entity annotation is encoded through the tag named *term* specified by its attributes *type* and *subtype*.

Figure 1 illustrates some of the mentioned encodings. The Georgian abbreviation tradition is especially complex and features many models, see Boeder (1987). Contraction, the mode of abbreviating by first and last letter which gained prominence in the Christian era, compare for instance Driscoll (2009) was very prominent in Old Georgian (abbreviation 1). According to (Danelia and Sarzhveladze, 2012, p.312), the following types of abbreviation are available in Old Georgian: the abbreviation of a word to its initial letter, suspension, contraction and elision of vowels. Suspension is very rare and only found on epigraphic monuments (it is not evidenced in manuscripts). Unlike manuscripts, in epigraphy often uncommon, unfamiliar abbreviations are present, which are difficult to decipher. When it came to suffixes, in Old Georgian affix chains are quite common. In order not to lose the meaning, the suffixes had to be encoded in the abbreviation and scribes may have had different opinions (apart from different spatial considerations) on how to extend the contraction principle consistently in this case (abbreviation

[3] `http://titus.uni-frankfurt.de/indexe.htm`: last accessed on 10.02.2017, `http://gnc.gov.ge/gnc/static/portal/gnc.html`,`http://museum.ge`

Abbreviation 1: ***k(rist')e***

```
<expan>
        <abbr>ქ</abbr>
        <ex>რისტ</ex>
        <abbr>ე</abbr>
</expan>
```

Abbreviation 2: ***k(rist')h(s)i***

```
<expan>
        <abbr>ქ</abbr>
        <ex>რისტ</ex>
        <abbr>ჩ</abbr>
        <ex>ს</ex>
        <abbr>ი</abbr>
</expan>
```

Abbreviation 3: ***k(rist')hsi***

```
<expan>
        <abbr>ქ</abbr>
        <ex>რისტ</ex>
        <abbr>ჩსი</abbr>
</expan>
```

Abbreviation 4: ***a(gh)m(a)***

```
<expan>
        <abbr>ა</abbr>
        <ex>ღ</ex>
        <abbr>მ</abbr>
        <ex>ა</ex>
</expan>
```

Named Entities: ***mepeta mepe davit***

```
<term type="namedEntity" subtype="title">მეფეთა მეფე</term>
<term type="namedEntity" subtype="anthroponym">დავით</term>
```

Filled Gap and Line Break: ***[va]r***

```
მე ვარ, რომელიც, თუ მე ვარ, ასრე გახდება, როგორც მე<lb n="8"/>
<supplied reason="lost">ვა</supplied>რ.
```

Figure 1: Examples of xml encodings in the corpus.

2 and 3). While contraction was especially important for named entities and in particular biblical individuals and places (nomina sacra), for other word classes other ways of abbreviation are found (abbreviation 4). We annotated titles such as king of kings (named entities) in order to relate inscription type to state organization and to better distinguish individuals of the same name.

Throughout the corpus one sees that inscriptions are fragmentary, some to the extent not to allow a full reconstruction of their texts. On average, an inscription had roughly 33 words, 4 gaps and 11 abbreviations.[4] Experts on language and inscriptions have been able to provide hypotheses about the full text of many inscriptions. However, many gaps remain. Not only for the already encoded inscriptions, but also for a planned extension to the corpus some computer-aided assistance in the reconstruction could be welcome. Since largely transcription and other epigraphic work is done in digital environments already, this paper asks: Can there be a tool assisting in reconstructing the complete texts of inscriptions? What will distinguish

[4] Not all abbreviations are counted here since some cannot be read or are concealed in undeciphered gaps.

such a tool from the traditional methods and resources such as lexica of abbreviations, lists of historical named entities and so forth.

2 Towards a Tool: Necessities

For reconstruction, a tool in the digital medium could be designed which assists in two important excerises of the epigrapher: expanding abbreviations and filling gaps. For this purpose, the text of the inscription could be represented digitally, where abbreviations and gaps could be marked and filled with precomputed guesses. However, machine leraning and related techniques have seemingly not yet been applied much to epigraphy, compare Bodel (2012). Some studies on abbreviations and word prediction in *psycholinguistics* may provide interesting and relevant insights eventhough they are not replicating the epigraphic context, see for instance Yang et al. (2009); McWilliam et al. (2009); Slattery et al. (2011); Taylor (1953).

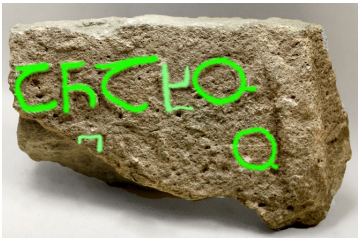
In computation, for tasks similar to epigraphic reconstruction such as *abbreviation generation*, *sequence prediction* and *spelling correction* feasible solutions have been found. But, those often rely on pretrained statistical models which need large amounts of input data. An example is the application of ngram language models for sequence prediction, where Manning and Schütze (1999) note that ngram models to be effective usually need large amounts of training data.[5] Even the full amount of Old Georgian data digitally available [6] is still not large enough to perform and thoroughly evaluate the majority of such approaches. Methodologically, there is an additional factor complicating assessment: Any gap can hold any number of abbreviations making gap filler generation (GFG) a more complex task than simple sequence prediction or abbreviation generation.

Additionally, the epigraphic record is very heterogeneous with the easier cases often already manually solved. In order to exemplify the heterogeneity of the epigraphic record and thus the range a tool aiding in reconstruction has to be able to address, we give some examples from the Georgian inscriptions, images come from the Corpus of Old Georgian Inscriptions.[7] On the one end there are inscriptions with so fragmentary evidence that no super computer can probably ever help to decipher the message, on the other there are reconstructions as trivial as to be performed without much effort even by laymen correctly.

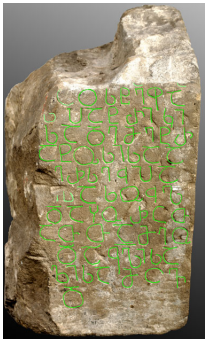
[5] See chapter 6 for discussion, (Manning and Schütze, 1999, p.201): "In general, four gram models do not become usable until one is training on several tens of millions of words of data."

[6] A subcorpus on Old Georgian not containing different redactions of the same texts from the TITUS server comprises roughly 4 million words.

[7] Collection under: `http://titus.fkidg1.uni-frankfurt.de/texte/etcg/cauc/ageo/inscr/carcera/carce.htm`



Although some letters survived, the extent and the placement of the gaps make a complete reconstruction almost impossible.



Here, the broken off part to the left can be reconstructed with a good level of confidence, since each line has more surviving than missing letters and since the amount of missing letters is of a minor magnitude. Also, there are few abbreviations.

Finally, in this example, only abbreviations of moderate difficulty have to be expanded, which could be done by a beginner to epigraphy knowing Old Georgian.

Facing such variety and difficulties, rather than provide a completed feasible solution for GFG, which given the scope of this article and the current landscape of computational epigraphic reconstruction would seem unrealistic, this paper primarily aims at making computational scholars aware of the inseparable interplay of abbreviation and gap which so characterizes the epigraphic record in many epochs, regions and languages and which may represent a new computational challenge. Towards a proof of concept however, a basic method for GFG is being formulated and tested.

In order to demonstrate the utility of such a tool, we concentrate on examples promising to yield some useful results. This is why we restrict ourselves for the time being to single words if possible not on broken off edges, the extent of which is unclear. We argue that if we are able to provide useful guesses for these, then larger units might be in reach for future research.

3 Method

We are looking for lexical matches of the gap context comparing two methods, pure frequency based cues and word embedding based cues. In the face of formulas and a very standardized language of inscriptions pure frequencies and conditional frequencies (of a word given a predecessor or follower) may be a sufficiently strong cue and could be feasible as a baseline. Word embeddings Mikolov et al. (2013a) on the other hand can be used for sequence prediciton since their training includes an optimization of immediate contextual similarity. To this end, semantic and syntactic similarity are captured by word embeddings which makes them a possible cue for a gap filler. Furthermore, Mikolov et al. (2013b) state that "neural network based language models significantly outperform N-gram models", compare Bengio et al. (2003); Mikolov et al. (2011); Schwenk (2007). In fact, in a pre-experiment, we found an ordinary n-gram language model to perform well only for the preditcion of the content of very short gaps. In order to generate word embeddings, we can later use for GFG, we compiled yet another corpus of Old Georgian texts fromt the TITUS archive.

3.1 Corpus

The TITUS website provides texts for many ancient languages and is (one of) the most comprehensive archive(s) (in close collaboration with the GNC) for Old Georgian text. Among the texts present are the Bible, lectionaries, hagiographical, theological and apocryphal texts, psalms and odes, song, historical texts, homiletic and exegetic texts, liturgical texts, canonical law texts, philosophical texts and for instance an astrological and a grammatical

text. Texts are often translated (from ancient Greek, Syriac and Armenian). For details, see the website.

These texts are thus of different genres than inscriptions, but the language stage is essentially the same. We extracted a subcorpus of roughly 4 million words, where for instance critical aparatuses or differing redactions have been omitted and only the critical text been taken. Punctuation as present has been separated from the tokens and tokens arranged so that sentences were approximately arranged in lines, as is usual for word embedding training. We do not entirely exclude the presence of noise. From the corpus, word2vec generated roughly 230,000 vectors for the wordforms in the corpus.

3.2 Approach

Each inscription was processed. An inscription internal gap was detected and the following mechanism tried to generate a filler.[8] First, the context of the gap was extracted. Here, ignoring any space within the gap(s), the continuous context to the left and right of the current gap until the next/previous space character has been extracted. If a subsequent gap was directly adjacent, there would be more than one gaps in such a "word". For instance same[bis]a[j] was so captured. Square brackets mark gaps, letters within are reconstructed, samebisaj means 'from Trinity'. This was then converted to a regex by simply substituting the letters of the gap by a placeholder: (same...a.).[9]

The regex was then used to match all candidates conforming to this pattern in the database of words of the Old Georgian corpus from which the word embeddings have also been generated.[10] [11] The outcome was a list of candidate fillers. However, depending on the extent and position of the gap, the number of fillers could easily become large. When one thinks of an aid for reconstruction, confronting the reconstructor with a large number of tokens, half of which is probably quite unlikely, will not be statisfactory. Therefore, we tried to use different cues for ranking candidates. Each candidate receives three values, firstly the cosine vector similarity to the word vector of the previous word if this word is in the lexicon (in the

[8] For the time being, gaps at the beginning or end of lines were left aside since their extent may be hard to estimate and validate, while the mechanism elaborated is under more based on gap breadth information.

[9] A more sophisticated approach would be to use the true breadth of the gap if annotated in absolute numbers. One could then assign a typical breadth to each letter and check if fillers are suitable for the gap at hand. A possible filler in its most condensed form should not be longer than the gap and its fully spelled out form not shorter. The way in which to generate the most condensed or gap matching form would pertain to abbreviation generation. One could for instance take the first letter of each word.

[10] For training, we used the default settings apart from the minCount feature which we set to 1 since the corpus is not huge and in this way, we capture hapax legomena and significantly enlarge embedding vocabulary.

[11] Neo4j was our data base system accessed via java.

Old Georgian corpus) and not gappy. Secondly, the same for the following word and thirdly, the filler is given its frequency from the Old Georgian corpus (in which it must occur since it has been extracted from there). From these values, we generate a weight for the candidates. This enables us then to sort the fillers and so limit the number of candidates to be offered to the reconstructor to a number he/she may deem useful. Such a number could be the top 10 for instance. However, since the weight may be the same for several candidates, we allow the limit to be exceeded and to include all candidates with a weight larger or equal to that of the tenth candidate.

3.3 Results

In the Old Georgian corpus, in overall 65 gappy "words" no fillers were retrieved for 25, whereas 26 of the 40 filler sets contained the correct filler. Results are encouraging, the correct filler was generated at a ratio of 0.65 decreasing to 0.6 if limiting the output to the top weights as described above using frequency as weighting cue. Recall was 0.62.[12] The average number of top fillers generated was roughly 7 which is not too confusable in terms of overview. Limiting to the top ranks had another effect, namely the Damerau Levenshtein distance, Damerau (1964) of the fillers to the correct solution decreased for more than half to be 4.21 which shows that even if the correct filler has not been included it is not unlikely to have a moderately similar or similar word in the top fillers. Using the word embedding cues, and only in case the previous and next word would both not be present in the word embedding lexicon frequency, deteriorated results. Taking the similarity to the last word if present (otherwise frequency) resulted in precision of 0.525, taking similarity to the next word if present (otherwise frequency) resulted in a precision of 0.5 and combinations such as the average of the similarities of last and next words if both were present, if only one of them was present that value and only in case none was present frequency, was still worse at 0.475. The correctly captured fillers from the embeddings however were largely coinciding but no subset of the ones captured by frequency.

3.4 Discussion and Post Experiment

Frequency is plainly connected with probability through bare counts, while word embeddings capture syntagmatic and paradigmatic similarity. Similarities to previous and following words performed at an almost equal level. One reason for the reduced performance in respect to frequency using only the immediately adjacent neighbours (the larger the context, the more probable the occurrence of a gap or abbreviation within the context) could be the

[12] Using fewer dimensions (10) only improved the result marginally in lowering the average rank at which the correct filler was to be found.

nature of language, namely the dichotomy between high frequency function words and content words. For the former, naturally many more neighbours exist in a training text which may make their vectors less specific and in turn less reliable ranking cues.

However, the amount of data tested on is not sufficient to conclude anything. Consequently, we tested the same method on 1,000 inscriptions of a Latin data base for inscriptions, the *Epigraphic Database Heidelberg.*[13] The text database, we used for computing word embeddings and extracting the frequency lexicon were the Latin Wikipedia[14] and the classical texts of the Packard Humanities Institute.[15] We found the same pattern as in Old Georgian, meanwhile with lower recall and precision. Frequency alone was the best cue. More research may shed light on the true reasons behind this pattern.

For the Latin dataset, another approach is feasible. A preliminary attempt is described and first results given in what follows as an outlook to future elaboration. Since there are more than 70,000 inscriptions, it makes sense to produce for instance 10 chunks of equal size (in terms of numbers of inscriptions). Then for gaps in any 1 chunk symbolizing the unreconstructed inscriptions, one can extract context and use pattern search in the 9 training chunks symbolizing the until then reconstructed inscriptions. Since inscriptions are highly stereotypical this may lead to good results. To test this assumption, in a small follow up on Latin, we extracted the context, this time regardless of spaces until the next/previous gap and then matched the resulting pattern *left_context.+right_context* from the inscriptions in the 9 held out chunks. The matches were checked for suitable length given the gap breadth. As described above, the most condensed form (each word abreviated by its first letter) should not be significantly longer and the fully spelled out form not significantly shorter than the space the gap offers. For each gap, we decreased context size by one character on each side and repeated matching until the context consisted in one character only. The matches (or fillers) were weighted for the length of the context at which they had been matched and for frequency of the match ($\sum_{i=1}^{n} |left\_context| + |right\_context|$ for n matches).

Here, we found a recall of 0.33 with the correct filler being present at a rate of 0.46 in the filler sets, whilst at a rate of 0.2 the correct filler was in the top 10 fillers. The average DL of the top fillers was 3.96 for those filler sets, where the correct match was not present. The highest ratios of correct matches per context lengths were achieved with longest contexts and balanced contexts, but length was a better cue than balance. To exemplify, a context of 5 characters to the left and 5 characters to the right is in total

[13] `http://edh-www.adw.uni-heidelberg.de/`
[14] `https://la.wikipedia.org`: last accessed on 16.12.2015
[15] `http://latin.packhum.org`: last accessed on 09.12.2015

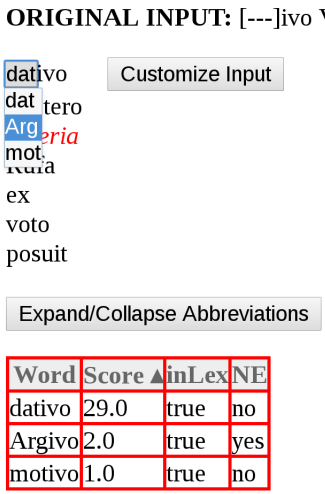


Figure 2: Simple front end example: The slightly transformed original transcription is visible in the first line. For each word, either the user is provided with a dropdown list restricted to the most probable automatically generated fillers or can choose to edit the gap filler manually. Abbreviations can be collapsed or expanded to support imagination of an original in the reconstructive process. A sortable table at the bottom informs him/her of all possibilities, which can be considerably more than in the thresheld dropdown menu and which contains additional information.

a 10 character context, but these contexts captured relatively less correct fillers than contexts of 0 characters to the left but 9 to the right. It seems that the longer a match in a continuous context, the better the cue.

4 User Interface

For the development of an "EpigraphyHelper" a user front end would have to be set-up. A sketch of this has been done using a platform independent HTML/Javascript solution which provides the most probable fillers in a drop-down container, see Figures 2 and 3. Future design and usability of this rendering should be made subject of an online survey for domain experts. The front end once finalized is completely independent from the technical backend, which is to say that the current method of generating gap fillers can be exchanged as soon as more effective methods are available.

The front end has several features. Firstly, the original transcription is presented on top, giving the epigrapher the context, he/she habitually encounters. Then per line each word is rendered either as non changeable text if readable as such on the inscription ('ex votu posuit' in the example) or

ORIGINAL INPUT: [P/ublio/] [M]ummio [P/ubli/] [f/ilio/] [Gal/eria/] [S]isenna[e] [Rutiliano] Xv[ir/o/] [stlitibus] [iudicandis] [---------]

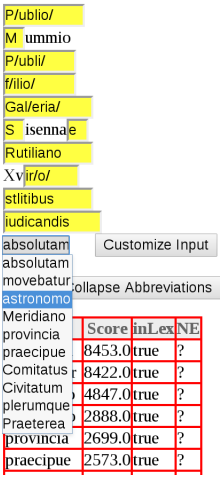


Figure 3: More complex example: Per word a separate line is assumed. Gaps filled by previous scientists as most probable reconstructions are editable. Visible and reconstructed abbreviations can be collapsed and expanded. They are marked differently.

with an expanded abbreviation, where the expansion is rendered in red and italics (Val*eria* in the example) or for each word which was reconstructed within a gap an editable textfield appears with yellow background, where abbreviations are marked by slashes (P/ublio/ in the example). Abbreviations can be collapsed and expanded per button. Finally, for gaps which have not been reconstructed, the algorithm computes candidates as described above and displays them in a drop-down list (Argivo in the example). Following Shneiderman's principle Shneiderman (1996), only in case of demand can the user obtain a sortable table with many more possibilities and additional annotations for the words. If none of the proposed fillers is deemed correct, the user can activate a 'Customize Input' button and transform the drop-down into an editable textfield.

5 Future Work and Experimentation

Of course, epigraphers have tried hard and succeeded well in reconstructions of inscriptions both internalizing abbreviation and text completion, connecting this with typical functional epigraphic formula and historical events and individuals. The frustration of not being able to decipher the message of certain inscriptions is probably a well known feeling for epigraphers and each one may have found his/her own way to deal with this issue. An application of AI to epigraphy should therefore not pretend to be a remedy for this frustration since it is clear that a too fragmentary inscription cannot be reasonably reconstructed. Yet, since the capacity of the human brain to keep in mind all relevant words, names and orthographic variants (and in consequence all possible reconstructions) is limited in comparison with a computer, a reconstruction aid may, in the best case find reasonable fillers for some of the not yet reconstructed gaps which had slipped the conscience of previous reconstructors. Especially in the case of Named Entities, a vast array of possibilities exists.

Furthermore, unreasonable candidates which such a system produces can be discarded by a human expert in a matter of seconds, leaving the technologically open user with a positive net outcome. One crucial question for an application of AI to epigraphy will be at which rate good guesses can be produced. Assessing such a question, databases such as the epigraphic database Heidelberg or the database Clauss/Slaby[16] may be seen as a benchmark dataset which will enable computer scientists to evaluate their approaches against the recontructions already conducted.

[16] `http://www.manfredclauss.de/`

6 Conclusion

A corpus of Old Georgian inscriptions has been compiled. Additionally, a tool for epigraphic reconstruction has been sketched in order to raise awareness in the Computer Scientific community that such a task exists, that data sets for its evaluation exist and that the task is an interesting computational challenge involving both abbreviation resolution or generation and sequence prediction. To this end, we have only been able to show that in the case of Old Georgian, thanks to a large resource of Old Georgian texts from the internet, a reconstruction aid can produce on average 7 fillers for roughly 60% of gaps with 60% of filler sets containing the correct solution. We hope for more general results and solutions in the future.

References

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bodel, J. (2012). Latin Epigraphy and the IT Revolution. *Proceedings of the British Academy*, 177:275 – 296.

Boeder, W. (1987). Versuch einer sprachwissenschaftlichen Interpretation der altgeorgischen Abkürzungen. *Revue des études géorgiennes et caucasiennes*, 3:33 – 81.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7:171–176.

Danelia, K. and Sarzhveladze, Z. (2012). *Kartuli p'aleograpia [Georgian Paleography]*. Nekeri.

Driscoll, M. (2009). Marking up abbreviations in old norse-icelandic manuscripts. In *Medieval Texts–Contemporary Media*. Ibis.

Gippert, J. (1995). Titus. das projekt eines indogermanistischen thesaurus ("titus. the project of an indo-european thesaurus"). *LDV-Forum*, 12(2):35–47.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

McWilliam, L., Schepman, A., and Rodway, P. (2009). The linguistic status of text message abbreviations: An exploration using a stroop task. *Computers in Human Behavior*, 25(4):970–974.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013b). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Deoras, A., Kombrink, S., Burget, L., and Černockỳ, J. (2011). Empirical evaluation and combination of advanced language modeling techniques. In *Twelfth Annual Conference of the International Speech Communication Association*.

Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language*, 21(3):492–518.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA. IEEE Computer Society.

Slattery, T. J., Schotter, E. R., Berry, R. W., and Rayner, K. (2011). Parafoveal and foveal processing of abbreviations during eye fixations in reading: making a case for case. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(4):1022.

Taylor, W. (1953). Cloze procedure: A new tool for measuring readability. *Journalism Quarterly*, 30:415–433.

Yang, D., Pan, Y.-c., and Furui, S. (2009). Automatic chinese abbreviation generation using conditional random field. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 273–276. Association for Computational Linguistics.