The effect of domain and text type on text prediction quality

Suzan Verberne, Antal van den Bosch, Helmer Strik, Lou Boves
Centre for Language Studies
Radboud University Nijmegen
s.verberne@let.ru.nl

Abstract

Text prediction is the task of suggesting text while the user is typing. Its main aim is to reduce the number of keystrokes that are needed to type a text. In this paper, we address the influence of text type and domain differences on text prediction quality.

By training and testing our text prediction algorithm on four different text types (Wikipedia, Twitter, transcriptions of conversational speech and FAQ) with equal corpus sizes, we found that there is a clear effect of text type on text prediction quality: training and testing on the same text type gave percentages of saved keystrokes between 27 and 34%; training on a different text type caused the scores to drop to percentages between 16 and 28%.

In our case study, we compared a number of training corpora for a specific data set for which training data is sparse: questions about neurological issues. We found that both text type and topic domain play a role in text prediction quality. The best performing training corpus was a set of medical pages from Wikipedia. The second-best result was obtained by leave-one-out experiments on the test questions, even though this training corpus was much smaller (2,672 words) than the other corpora (1.5 Million words).

1 Introduction

Text prediction is the task of suggesting text while the user is typing. Its main aim is to reduce the number of keystrokes that are needed to type a text, thereby saving time. Text prediction algorithms have been implemented for mobile devices, office software (Open Office Writer), search engines (Google query completion), and in special-needs software for writers who have difficulties typing (Garay-Vitoria and Abascal, 2006). In most applications, the scope of the prediction is the completion of the current word; hence the often-used term 'word completion'.

The most basic method for word completion is checking after each typed character whether the prefix typed since the last whitespace is unique according to a lexicon. If it is, the algorithm suggests to complete the prefix with the lexicon entry. The algorithm may also suggest to complete a prefix even before the word's uniqueness point is reached, using statistical information on the previous context. Moreover, it has been shown that significantly better prediction results can be obtained if not only the prefix of the current word is included as previous context, but also previous words (Fazly and Hirst, 2003) or characters (Van den Bosch and Bogers, 2008).

In the current paper, we follow up on this work by addressing the influence of text type and domain differences on text prediction quality. Brief messages on mobile devices (such as text messages, Twitter and Facebook updates) are of a different style and lexicon than documents typed in office software (Westman and Freund, 2010). In addition, the topic domain of the text also influences its content. These differences may cause an algorithm trained on one text type or domain to perform poorly on another.

The questions that we aim to answer in this paper are (1) "What is the effect of text type differences on the quality of a text prediction algorithm?" and (2) "What is the best choice of training data if domain- and text type-specific data is sparse?". To answer these questions, we perform three experiments:

1. A series of within-text type experiments on four different types of Dutch text: Wikipedia articles, Twitter data, transcriptions of con-

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 561–569,
Avignon, France, April 23 - 27 2012.

versational speech and web pages of Frequently Asked Questions (FAQ).

2. A series of across-text type experiments in which we train and test on different text types;
3. A case study using texts from a specific domain and text type: questions about neurological issues. Training data for this combination of language (Dutch), text type (FAQ) and domain (medical/neurological) is sparse. Therefore, we search for the type of training data that gives the best prediction results for this corpus. We compare the following training corpora:
    - The corpora that we compared in the text type experiments: Wikipedia, Twitter, Speech and FAQ, 1.5 Million words per corpus.
    - A 1.5 Million words training corpus that is of the same domain as the target data: medical pages from Wikipedia;
    - The 359 questions from the neuro-QA data themselves, evaluated in a leave-one-out setting (359 times training on 358 questions and evaluating on the remaining questions).

The prospective application of the third series of experiments is the development of a text prediction algorithm in an online care platform: an online community for patients seeking information about their illness. In this specific case the target group is patients with language disabilities due to neurological disorders.

The remainder of this paper is organized as follows: In Section 2 we give a brief overview of text prediction methods discussed in the literature. In Section 3 we present our approach to text prediction. Sections 4 and 5 describe the experiments that we carried out and the results we obtained. We phrase our conclusions in Section 6.

2 Text prediction methods

Text prediction methods have been developed for several different purposes. The older algorithms were built as communicative devices for people with disabilities, such as motor and speech impairments. More recently, text prediction is developed for writing with reduced keyboards, specifically for writing (composing messages) on mobile devices (Garay-Vitoria and Abascal, 2006).

All modern methods share the general idea that previous context (which we will call the 'buffer') can be used to predict the next block of characters (the 'predictive unit'). If the user gets correct suggestions for continuation of the text then the number of keystrokes needed to type the text is reduced. The unit to be predicted by a text prediction algorithm can be anything ranging from a single character (which actually does not save any keystrokes) to multiple words. Single words are the most widely used as prediction units because they are recognizable at a low cognitive load for the user, and word prediction gives good results in terms of keystroke savings (Garay-Vitoria and Abascal, 2006).

There is some variation among methods in the size and type of buffer used. Most methods use character n-grams as buffer, because they are powerful and can be implemented independently of the target language (Carlberger, 1997). In many algorithms the buffer is cleared at the start of each new word (making the buffer never larger than the length of the current word). In the paper by (Van den Bosch and Bogers, 2008), two extensions to the basic prefix-model are compared. They found that an algorithm that uses the previous n characters as buffer, crossing word borders without clearing the buffer, performs better than both a prefix character model and an algorithm that includes the full previous word as feature. In addition to using the previously typed characters and/or words in the buffer, word characteristics such as frequency and recency could also be taken into account (Garay-Vitoria and Abascal, 2006).

Possible evaluation measures for text prediction are the proportion of words that are correctly predicted, the percentage of keystrokes that could maximally be saved (if the user would always make the correct decision), and the time saved by the use of the algorithm (Garay-Vitoria and Abascal, 2006). The performance that can be obtained by text prediction algorithms depends on the language they are evaluated on. Lower results are obtained for higher-inflected languages such as German than for low-inflected languages such as English (Matiasek et al., 2002). In their overview of text prediction systems, (Garay-Vitoria and Abascal, 2006) report performance scores ranging from 29% to 56% of keystrokes saved.

An important factor that is known to influence the quality of text prediction systems, is training

set size (Lesher et al., 1999; Van den Bosch, 2011). The paper by (Van den Bosch, 2011) shows log-linear learning curves for word prediction (a constant improvement each time the training corpus size is doubled), when the training set size is increased incrementally from 10^2 to $3*10^7$ words.

3 Our approach to text prediction

We implement a text prediction algorithm for Dutch, which is a productive compounding language like German, but has a somewhat simpler inflectional system. We do not focus on the effect of training set size, but on the effect of text type and topic domain differences.

Our approach to text prediction is largely inspired by (Van den Bosch and Bogers, 2008). We experiment with two different buffer types that are based on character n-grams:

- 'Prefix of current word' contains all characters of only the word currently keyed in, where the buffer shifts by one character position with every new character.
- 'Buffer15' buffer also includes any other characters keyed in belonging to previously keyed-in words.

Modeling character history beyond the current word can naturally be done with a buffer model in which the buffer shifts by one position per character, while a typical left-aligned prefix model (that never shifts and fixes letters to their positional feature) would not be able to do this.

In the buffer, all characters from the text are kept, including whitespace and punctuation. The predictive unit is one token (word or punctuation symbol). In both the buffer and the prediction label, any capitalization is kept. At each point in the typing process, our algorithm gives one suggestion: the word that is the most likely continuation of the current buffer.

We save the training data as a classification data set: each character in the buffer fills a feature slot and the word that is to be predicted is the classification label. Figures 1 and 2 give examples of each of the buffer types Prefix and Buffer15 that we created for the text fragment "*tot een niveau*" in the context "stelselmatig bij elke verkiezing tot een niveau van' '(*structurally with each election to a level of*). We use the implementation of the IGTree decision tree algorithm in TiMBL (Daelemans et al., 1997) to train our models.

3.1 Evaluation

We evaluate our algorithms on corpus data. This means that we have to make assumptions about user behaviour. We assume that the user confirms a suggested word as soon as it is suggested correctly, not typing any additional characters before confirming. We evaluate our text prediction algorithms in terms of the percentage of keystrokes saved K:

$$K = \frac{\sum_{i=0}^{n}(F_i) - \sum_{i=0}^{n}(W_i)}{\sum_{i=0}^{n}(F_i)} * 100 \qquad (1)$$

in which n is the number of words in the test set, W_i is the number of keystrokes that have been typed before the word i is correctly suggested and F_i is the number of keystrokes that would be needed to type the complete word i. For example, our algorithm correctly predicts the word *niveau* after the context `i n g _ t o t _ e e n _ n i v` in the test set. Assuming that the user confirms the word *niveau* at this point, three keystrokes were needed for the prefix *niv*. So, $W_i = 3$ and $F_i = 6$. The number of keystrokes needed for whitespace and punctuation are unchanged: these have to be typed anyway, independently of the support by a text prediction algorithm.

4 Text type experiments

In this section, we describe the first and second series of experiments. The case study on questions from the neurological domain is described in Section 5.

4.1 Data

In the text type experiments, we evaluate our text prediction algorithm on four different types of Dutch text: Wikipedia, Twitter data, transcriptions of conversational speech, and web pages of Frequently Asked Questions (FAQ). The Wikipedia corpus that we use is part of the Lassy corpus (Van Noord, 2009); we obtained a version from the summer of 2010.[1] The Twitter data are collected continuously and automatically filtered for language by Erik Tjong Kim Sang (Tjong Kim Sang, 2011). We used the tweets from all users that posted at least 19 tweets (excluding retweets) during one day in June 2011. This is a set of 1 Million Twitter messages from 30,000

[1] http://www.let.rug.nl/vannoord/trees/Treebank/Machine/NLWIKI20100826/COMPACT/

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	_	t	tot
-	-	-	-	-	-	-	-	-	-	-	-	-	-	_	t	o	tot
-	-	-	-	-	-	-	-	-	-	-	-	-	-	t	o	t	tot
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	_	e	een
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	e	een
-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	e	n	een
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	_	n	niveau
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	n	i	niveau
-	-	-	-	-	-	-	-	-	-	-	-	-	-	n	i	v	niveau
-	-	-	-	-	-	-	-	-	-	-	-	-	n	i	v	e	niveau
-	-	-	-	-	-	-	-	-	-	-	-	n	i	v	e	a	niveau
-	-	-	-	-	-	-	-	-	-	-	n	i	v	e	a	u	niveau

Figure 1: Example of buffer type 'Prefix' for the text fragment "(elke verkiezing) tot een niveau". Underscores represent whitespaces.

l	k	e	_	v	e	r	k	i	e	z	i	n	g	_	tot
k	e	_	v	e	r	k	i	e	z	i	n	g	_	t	tot
e	_	v	e	r	k	i	e	z	i	n	g	_	t	o	tot
_	v	e	r	k	i	e	z	i	n	g	_	t	o	t	tot
v	e	r	k	i	e	z	i	n	g	_	t	o	t	_	een
e	r	k	i	e	z	i	n	g	_	t	o	t	_	e	een
r	k	i	e	z	i	n	g	_	t	o	t	_	e	e	een
k	i	e	z	i	n	g	_	t	o	t	_	e	e	n	een
i	e	z	i	n	g	_	t	o	t	_	e	e	n	_	niveau
e	z	i	n	g	_	t	o	t	_	e	e	n	_	n	niveau
z	i	n	g	_	t	o	t	_	e	e	n	_	n	i	niveau
i	n	g	_	t	o	t	_	e	e	n	_	n	i	v	niveau
n	g	_	t	o	t	_	e	e	n	_	n	i	v	e	niveau
g	_	t	o	t	_	e	e	n	_	n	i	v	e	a	niveau
_	t	o	t	_	e	e	n	_	n	i	v	e	a	u	niveau

Figure 2: Example of buffer type 'Buffer15' for the text fragment "(elke verkiezing) tot een niveau". Underscores represent whitespaces.

different users. The transcriptions of conversational speech are from the Spoken Dutch Corpus (CGN) (Oostdijk, 2000); for our experiments, we only use the category 'spontaneous speech'. We obtained the FAQ data by downloading the first 1,000 pages that Google returns for the query 'faq' with the language restriction Dutch. After cleaning the pages from HTML and other coding, the resulting corpus contained approximately 1.7 Million words of questions and answers.

4.2 Within-text type experiments

For each of the four text types, we compare the buffer types 'Prefix' and 'Buffer15'. In each experiment, we use 1.5 Million words from the corpus to train the algorithm and 100,000 words to test it. The results are in Table 1.

4.3 Across-text type experiments

We investigate the importance of text type differences for text prediction with a series of experiments in which we train and test our algorithm on texts of different text types. We keep the size of the train and test sets the same: 1.5 Million words and 100,000 words respectively. The results are in Table 2.

4.4 Discussion of the results

Table 1 shows that for all text types, the buffer of 15 characters that crosses word borders gives better results than the prefix of the current word only. We get a relative improvement of 35% (for FAQ) to 62% (for Speech) of Buffer15 compared to Prefix-only.

Table 2 shows that text type differences have an influence on text prediction quality: all across-text type experiments lead to lower results than the within-text type experiments. From the results in Table 2, we can deduce that of the four text types, speech and Twitter language resemble each other more than they resemble the other two, and Wikipedia and FAQ resemble each other more. Twitter and Wikipedia data are the least similar: training on Wikipedia data makes the text prediction score for Twitter data drop from 29.2 to 16.5%.[2]

[2]Note that the results are not symmetric. For example,

Table 1: Results from the within-text type experiments in terms of percentages of saved keystrokes. *Prefix* means: 'use the previous characters of the current word as features'. *Buffer 15* means 'use a buffer of the previous 15 characters as features'.

	Prefix	Buffer15
Wikipedia	22.2%	30.5%
Twitter	21.3%	29.2%
Speech	20.7%	33.4%
FAQ	20.2%	27.2%

Table 2: Results from the across-text type experiments in terms of percentages of saved keystrokes, using the best-scoring configuration from the within-text type experiments: a buffer of 15 characters

Trained on	Tested on Wikipedia	Tested on Twitter	Tested on Speech	Tested on FAQ
Wikipedia	30.5%	16.5%	22.3%	24.9%
Twitter	17.9%	29.2%	27.9%	20.7%
Speech	19.7%	22.5%	33.4%	21.0%
FAQ	22.6%	18.2%	22.9%	27.2%

5 Case study: questions about neurological issues

Online care platforms aim to bring together patients and experts. Through this medium, patients can find information about their illness, and get in contact with fellow-sufferers. Patients who suffer from neurological damage may have communicative disabilities because their speaking and writing skills are impaired. For these patients, existing online care platforms are often not easily accessible. Aphasia, for example, hampers the exchange of information because the patient has problems with word finding.

In the project 'Communicatie en revalidatie DigiPoli' (ComPoli), language and speech technologies are implemented in the infrastructure of an existing online care platform in order to facilitate communication for patients suffering from neurological damage. Part of the online care platform is a list of frequently asked questions about neurological diseases with answers. A user can browse through the questions using a chat-by-click interface (Geuze et al., 2008). Besides reading the listed questions and answers, the user has the option to submit a question that is not yet included in the list. The newly submitted questions are sent to an expert who answers them and adds both question and answer to the chat-by-click database. In typing the question to be submitted, the user will be supported by a text prediction application.

The aim of this section is to find the best training corpus for newly formulated questions in the neurological domain. We realize that questions formulated by users of a web interface are different from questions formulated by experts for the purpose of a FAQ-list. Therefore, we plan to gather real user data once we have a first version of the user interface running online. For developing the text prediction algorithm that is behind the initial version of the application, we aim to find the best training corpus using the questions from the chat-by-click data as training set.

5.1 Data

The chat-by-click data set on neurological issues consists of 639 questions with corresponding answers. A small sample of the data (translated to English) is shown in Table 3. In order to create the test data for our experiments, we removed duplicate questions from the chat-by-click data, leaving a set of 359 questions.[3]

In the previous sections, we used corpora of 100,000 words as test collections and we calculated the percentage of saved keystrokes over the

training on Wikipedia, testing on Twitter gives a different result from training on Twitter, testing on Wikipedia. This is due to the size and domain of the vocabularies in both data sets and the richness of the contexts (in order for the algorithm to predict a word, it has to have seen it in the train set). If the test set has a larger vocabulary than the train set, a lower proportion of words can be predicted than when it is the other way around.

[3]Some questions and answers are repeated several times in the chat-by-click data because they are located at different places in the chat-by-click hierarchy.

Table 3: A sample of the neuro-QA data, translated to English.

question_0_505	Can (P)LS be cured?
answer_0_505	Unfortunately, a real cure is not possible. However, things can be done to combat the effects of the diseases, mainly relieving symptoms such as stiffness and spasticity. The phisical therapist and rehabilitation specialist can play a major role in symptom relief. Moreover, there are medications that can reduce spasticity.
question_0_508	How is (P)LS diagnosed?
answer_0_508	The diagnosis PLS is difficult to establish, especially because the symptoms strongly resemble HSP symptoms (Strumpell's disease). Apart from blood and muscle research, several neurological examinations will be carried out.

Table 4: Results for the neuro-QA questions only in terms of percentages of saved keystrokes, using different training sets. The text prediction configuration used in all settings is Buffer15. The test samples are 359 questions with an average length of 7.5 words. The percentages of saved keystrokes are means over the 359 questions.

Training corpus	# words	Mean % of saved keystrokes in neuro-QA questions (stdev)	OOV-rate
Twitter	1.5 Million	13.3% (12.5)	28.5%
Speech	1.5 Million	14.1% (13.2)	26.6%
Wikipedia	1.5 Million	16.1% (13.1)	19.4%
FAQ	1.5 Million	19.4% (15.6)	20.0%
Medical Wikipedia	1.5 Million	28.1% (16.5)	7.0%
Neuro-QA questions (leave-one-out)	2,672	26.5% (19.9)	17.8%

complete test corpus. In the reality of our case study however, users will type only brief fragments of text: the length of the question they want to submit. This means that there is potentially a large deviation in the effectiveness of the text prediction algorithm per user, depending on the content of the small text they are typing. Therefore, we decided to evaluate our training corpora separately on each of the 359 unique questions, so that we can report both mean and standard deviation of the text prediction scores on small (realistically sized) samples. The average number of words per question is 7.5; the total size of the neuro-QA corpus is 2,672 words.

5.2 Experiments

We aim to find the training set that gives the best text prediction result for the neuro-QA questions. We compare the following training corpora:

- The corpora that we compared in the text type experiments: Wikipedia, Twitter, Speech and FAQ, 1.5 Million words per corpus.
- A 1.5 Million words training corpus that is of the same topic domain as the target data: Wikipedia articles from the medical domain;
- The 359 questions from the neuro-QA data themselves, evaluated in a leave-one-out setting (359 times training on 358 questions and evaluating on the remaining questions).

In order to create the 'medical Wikipedia' corpus, we consulted the category structure of the Wikipedia corpus. The Wikipedia category 'Geneeskunde' (*Medicine*) contains 69,898 pages and in the deeper nodes of the hierarchy we see many non-medical pages, such as trappist beers (ordered under beer, booze, alcohol, Psychoactive drug, drug, and then medicine). If we remove all pages that are more than five levels under the 'Geneeskunde' category root, 21,071 pages are left, which contain fairly over the 1.5 Million words that we need. We used the first 1.5 Million words of the corpus in our experiments.

The text prediction results for the different corpora are in Table 4. For each corpus, the out-of-vocabulary rate is given: the percentage of words in the Neuro-QA questions that do not occur in the corpus.[4]

5.3 Discussion of the results

We measured the statistical significance of the mean differences between all text prediction scores using a Wilcoxon Signed Rank test on paired results for the 359 questions. We found that

[4]The OOV-rate for the Neuro-QA corpus itself is the average of the OOV-rate of each leave-one-out experiment: the proportion of words that only occur in one question.

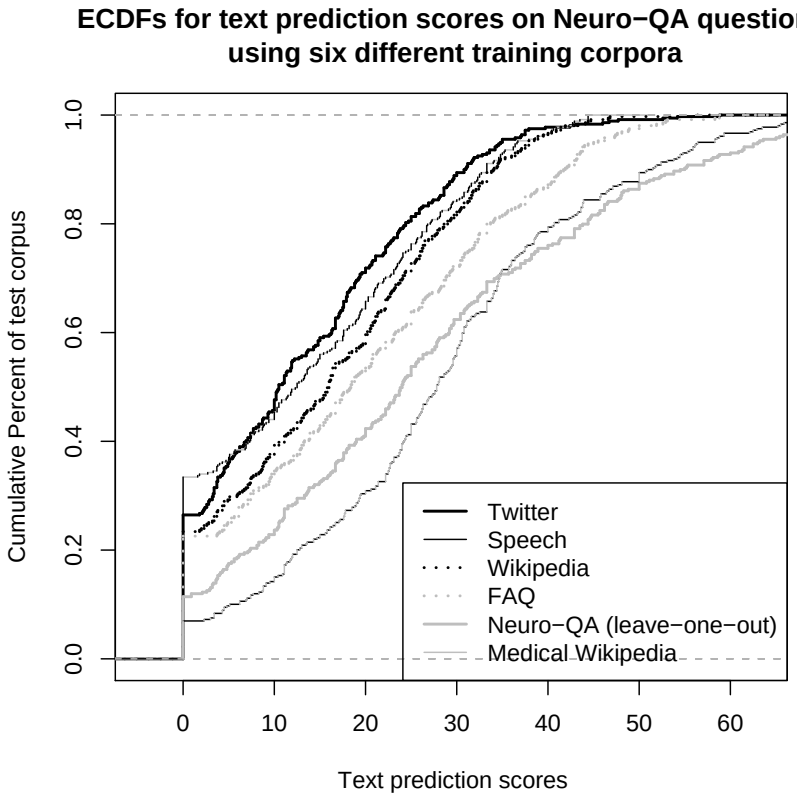

Figure 3: Empirical CDFs for text prediction scores on Neuro-QA data. Note that the curves that are at the bottom-right side represent the better-performing settings.

the difference between the Twitter and Speech corpora on the task is not significant ($P = 0.18$). The difference between Neuro-QA and Medical Wikipedia is significant with $P = 0.02$; all other differences are significant with $P < 0.01$.

The Medical Wikipedia corpus and the leave-one-out experiments on the Neuro-QA data give better text prediction scores than the other corpora. The Medical Wikipedia even scores slightly better than the Neuro-QA data itself. Twitter and Speech are the least-suited training corpora for the Neuro-QA questions, and FAQ data gives a bit better results than a general Wikipedia corpus.

These results suggest that both text type and topic domain play a role in text prediction quality, but the high scores for the Medical Wikipedia corpus shows that topic domain is even more important than text type.[5] The column 'OOV-rate' shows that this is probably due to the high coverage of terms in the Neuro-QA data by the Medical Wikipedia corpus.

Table 4 also shows that the standard deviation among the 359 samples is relatively large. For some questions, we 0% of the keystrokes are saved, while for other, scores of over 80% are obtained (by the Neuro-QA and Medical Wikipedia training corpora). We further analyzed the differences between the training sets by plotting the Empirical Cumulative Distribution Function (ECDF) for each experiment. An ECDF shows the development of text prediction scores (shown on the X-axis) by walking through the test set in 359 steps (shown on the Y-axis).

The ECDFs for our training corpora are in Figure 3. Note that the curves that are at the bottom-right side represent the better-performing settings (they get to a higher maximum after having seen a smaller portion of the samples). From Figure 3, it is again clear that the Neuro-QA and Medical Wikipedia corpora outperform the other training corpora, and that of the other four, FAQ is the best-performing corpus. Figure 3 also shows a large difference in the sizes of the starting percentiles: The proportion of samples with a text prediction

[5]We should note here that we did not control for domain differences between the four different text types. They are intended to be 'general domain' but Wikipedia articles will naturally be of different topics than conversational speech.

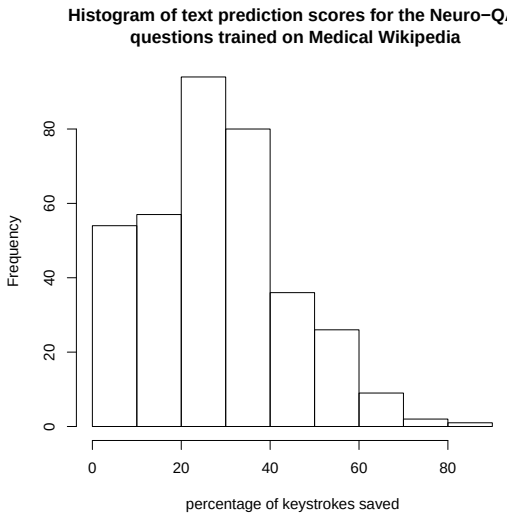

Figure 4: Histogram of text prediction scores for the Neuro-QA questions trained on Medical Wikipedia. Each bin represents 36 questions.

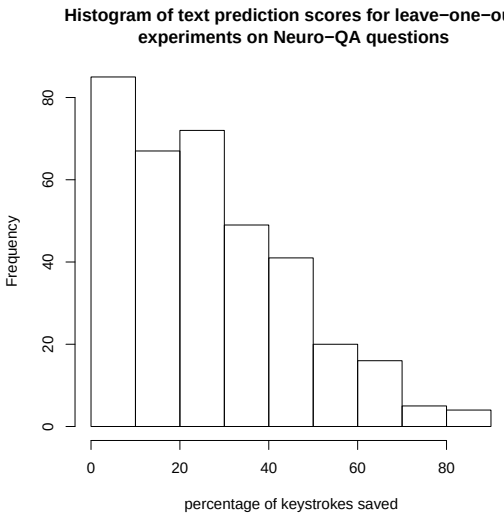

Figure 5: Histogram of text prediction scores for leave-one-out experiments on Neuro-QA questions. Each bin represents 36 questions.

score of 0% is less than 10% for the Medical Wikipedia up to more than 30% for Speech.

We inspected the questions that get a text prediction score of 0%. We see many medical terms in these questions, and many of the utterances are not even questions, but multi-word terms representing topical headers in the chat-by-click data. Seven samples get a zero-score in the output of all six training corpora, e.g.:

- glycogenose III.
- potassium-aggrevated myotonias.

26 samples get a zero-score in the output of all training corpora except for Medical Wikipedia and Neuro-QA itself. These are mainly short headings with domain-specific terms such as:

- idiopatische neuralgische amyotrofie.
- Markesbery-Griggs distale myopathie.
- oculopharyngeale spierdystrofie.

Interestingly, the ECDFs show that the Medical Wikipedia and Neuro-QA corpora cross at around percentile 70 (around the point of 40% saved keystrokes). This indicates that although the means of the two result samples are close to each other, the distribution the scores for the individual questions is different. The histograms of both distributions (Figures 4 and 5) confirm this: the algorithm trained on the Medical Wikipedia corpus leads a larger number of samples with scores around the mean, while the leave-one-out experiments lead to a larger number of samples with low prediction scores and a larger number of samples with high prediction scores. This is also reflected by the higher standard deviation for Neuro-QA than for Medical Wikipedia.

Since both the leave-one-out training on the Neuro-QA questions and the Medical Wikipedia led to good results but behave differently for different portions of the test data, we also evaluated a combination of both corpora on our test set: We created training corpora consisting of the Medical Wikipedia corpus, complemented by 90% of the Neuro-QA questions, testing on the remaining 10% of the Neuro-QA questions. This led to mean percentage of saved keystrokes of 28.6%, not significantly higher than just the Medical Wikipedia corpus.

6 Conclusions

In Section 1, we asked two questions: (1) "What is the effect of text type differences on the quality of a text prediction algorithm?" and (2) "What is the best choice of training data if domain- and text type-specific data is sparse?"

By training and testing our text prediction algorithm on four different text types (Wikipedia, Twitter, transcriptions of conversational speech and FAQ) with equal corpus sizes, we found that there is a clear effect of text type on text prediction quality: training and testing on the same text type

gave percentages of saved keystrokes between 27 and 34%; training on a different text type caused the scores to drop to percentages between 16 and 28%.

In our case study, we compared a number of training corpora for a specific data set for which training data is sparse: questions about neurological issues. We found significant differences between the text prediction scores obtained with the six training corpora: the Twitter and Speech corpora were the least suited, followed by the Wikipedia and FAQ corpus. The highest scores were obtained by training the algorithm on the medical pages from Wikipedia, immediately followed by leave-one-out experiments on the 359 neurological questions. The large differences between the lexical coverage of the medical domain played a central role in the scores for the different training corpora.

Because we obtained good results by both the Medical Wikipedia corpus and the neuro-QA questions themselves, we opted for a combination of both data types as training corpus in the initial version of the online text prediction application. Currently, a demonstration version of the application is running for ComPoli-users. We hope to collect questions from these users to re-train our algorithm with more representative examples.

Acknowledgments

This work is part of the research programme 'Communicatie en revalidatie digiPoli' (ComPoli[6]), which is funded by ZonMW, the Netherlands organisation for health research and development.

References

J. Carlberger. 1997. Design and Implementation of a Probabilistic Word Prediciton Program. Master thesis, Royal Institute of Technology (KTH), Sweden.

W. Daelemans, A. Van Den Bosch, and T. Weijters. 1997. IGTree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11(1):407–423.

A. Fazly and G. Hirst. 2003. Testing the efficacy of part-of-speech information in word completion. In *Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods*, pages 9–16.

N. Garay-Vitoria and J. Abascal. 2006. Text prediction systems: a survey. *Universal Access in the Information Society*, 4(3):188–203.

J. Geuze, P. Desain, and J. Ringelberg. 2008. Rephrase: chat-by-click: a fundamental new mode of human communication over the internet. In *CHI'08 extended abstracts on Human factors in computing systems*, pages 3345–3350. ACM.

G.W. Lesher, B.J. Moulton, D.J. Higginbotham, et al. 1999. Effects of ngram order and training text size on word prediction. In *Proceedings of the RESNA '99 Annual Conference*, pages 52–54.

Johannes Matiasek, Marco Baroni, and Harald Trost. 2002. FASTY - A Multi-lingual Approach to Text Prediction. In Klaus Miesenberger, Joachim Klaus, and Wolfgang Zagler, editors, *Computers Helping People with Special Needs*, volume 2398 of *Lecture Notes in Computer Science*, pages 165–176. Springer Berlin / Heidelberg.

N. Oostdijk. 2000. The spoken Dutch corpus: overview and first evaluation. In *Proceedings of LREC-2000, Athens*, volume 2, pages 887–894.

Erik Tjong Kim Sang. 2011. Het gebruik van Twitter voor Taalkundig Onderzoek. In *TABU: Bulletin voor Taalwetenschap*, volume 39, pages 62–72. In Dutch.

A. Van den Bosch and T. Bogers. 2008. Efficient context-sensitive word completion for mobile devices. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 465–470. ACM.

A. Van den Bosch. 2011. Effects of context and recency in scaled word completion. *Computational Linguistics in the Netherlands Journal*, 1:79–94, 12/2011.

G. Van Noord. 2009. Huge parsed corpora in LASSY. In *Proceedings of The 7th International Workshop on Treebanks and Linguistic Theories (TLT7)*.

S. Westman and L. Freund. 2010. Information Interaction in 140 Characters or Less: Genres on Twitter. In *Proceedings of the third symposium on Information Interaction in Context (IIiX)*, pages 323–328. ACM.

[6] http://lands.let.ru.nl/~strik/research/ComPoli/