Clinical Text Prediction with Numerically Grounded Conditional Language Models

Georgios P. Spithourakis
Department of Computer Science
University College London
g.spithourakis@cs.ucl.ac.uk

Steffen E. Petersen
William Harvey Research Institute
Queen Mary University of London
s.e.petersen@qmul.ac.uk

Sebastian Riedel
Department of Computer Science
University College London
s.riedel@cs.ucl.ac.uk

Abstract

Assisted text input techniques can save time and effort and improve text quality. In this paper, we investigate how grounded and conditional extensions to standard neural language models can bring improvements in the tasks of word prediction and completion. These extensions incorporate a structured knowledge base and numerical values from the text into the context used to predict the next word. Our automated evaluation on a clinical dataset shows extended models significantly outperform standard models. Our best system uses both conditioning and grounding, because of their orthogonal benefits. For word prediction with a list of 5 suggestions, it improves recall from 25.03% to 71.28% and for word completion it improves keystroke savings from 34.35% to 44.81%, where theoretical bound for this dataset is 58.78%. We also perform a qualitative investigation of how models with lower perplexity occasionally fare better at the tasks. We found that at test time numbers have more influence on the document level than on individual word probabilities.

1 Introduction

Text prediction is the task of suggesting the next word, phrase or sentence while the user is typing. It is an assisted data entry function that aims to save time and effort by reducing the number of keystrokes needed and to improve text quality by preventing misspellings, promoting adoption of standard terminologies and allowing for exploration of the vocabulary (Sevenster and Aleksovski, 2010; Sevenster et al., 2012).

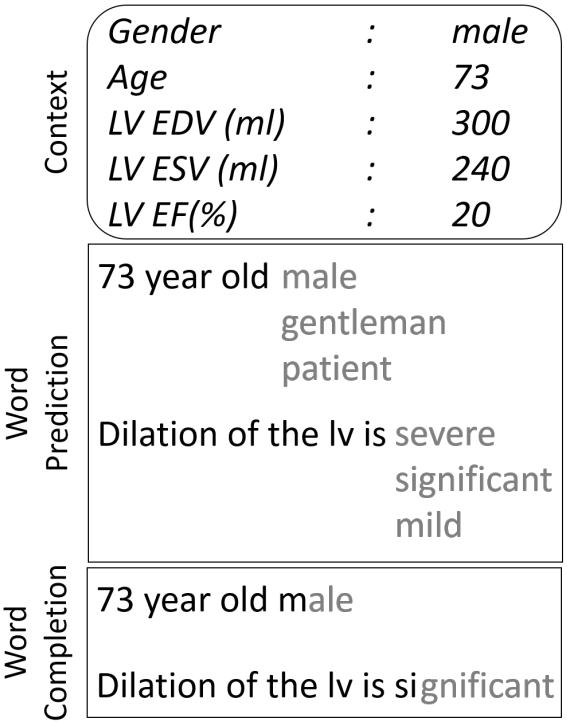

Figure 1: Word prediction and completion tasks. A system makes suggestions (in grey) for the next word and to complete a word as it is being typed, respectively. The context is often relevant to the quality of the suggestions.

Text prediction originated in augmentative and alternative communication (AAC) to increase text generation rates for people with motor or speech impairments (Beukelman and Mirenda, 2005). Its scope has been extended to a gamut of applications, such as data entry in mobile devices (Dunlop and Crossan, 2000), interactive machine translation (Foster et al., 2002), search term auto-

Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 6–16,
Austin, TX, November 5, 2016.

completion (Bast and Weber, 2006) and assisted clinical report compilation (Eng and Eisner, 2004; Cannataro et al., 2012).

In this paper, we explore the tasks of word prediction, where a system displays a list of suggestions for the next word before the user starts typing it, and word completion, where the system suggests a single possible completion for the word, while the user is typing its characters. The former task is relevant when the user has not yet made a firm decision about the intended word, thus any suggestions can have a great impact in the content of the final document. In the latter case, the user is thinking of a particular word that they want to input and the system's goal is to help them complete the word as quickly as possible. Figure 1 shows examples for both tasks.

Often, the user's goal is to compose a document describing a particular situation, e.g. a clinical report about a patient's condition. An intelligent predictive system should be able to account for such contextual information in order to improve the quality of its suggestions. Challenges to modelling structured contexts include mixed types of values for the different fields an schema inconsistencies across the entries of the structure. We address these issues by employing numerically grounded conditional language models (Spithourakis et al., 2016).

The contribution of this work is twofold. First, we show that conditional and numerically grounded models can achieve significant improvements over standard language models in the tasks of word prediction and completion. Our best model with a list of 5 suggestions raises recall from 25.03% to 71.28% and keystroke savings from 34.35% to 44.81%. Second, we investigate in depth the behaviour of such models and their sensitivity to the numerical values in the text. We find that the grounded probability for the whole document is more sensitive to numerical configurations than the probabilities of individual words.

2 Related Work

There have been several applications of text prediction systems in the clinical domain. Word completion has been a feature of discharge summary (Chen et al., 2012), brain MRI report (Cannataro et al., 2012) and radiology report (Eng and Eisner, 2004) compilation systems. Aiming towards clinical document standardisation, Sirel (2012) adopted the ICD-10 medical classification codes as a lexical resource and Lin et al. (2014) built a semi-automatic annotation tool to generate entry-level interoperable clinical documents.

Hua et al. (2014) reported 13.0% time reduction and 3.9% increase of response accuracy in a data entry task. Gong et al. (2016) found a performance of 87.1% for keystroke savings, a 70.5% increase in text generation rate, a 34.1% increase in reporting comprehensiveness and a 14.5% reduction in non-adherence to fields when reporting on patient safety event. In non-clinical applications, a survey of text prediction systems (Garay-Vitoria and Abascal, 2006) reports keystroke savings ranging from 29% to 56%.

The context provided to the predictive system can have a significant effect on its performance. Fazly and Hirst (2003) and Van Den Bosch and Bogers (2008) obtained significantly better results for word completion by considering not only the prefix of the current word but also previous words and characters, respectively. Wandmacher and Antoine (2008) explored methods to integrate n-gram language models with semantic information and Trnka (2008) used topic-adapted language models for word prediction. More recently, Ghosh et al. (2016) incorporated sentence topics as contextual features into a neural language model and reported perplexity improvements in a word prediction task. None of these systems considers structured background information or numerical values from the text as additional context.

The motivation to include this information as context to text prediction system is based on the importance of numerical quantities to textual entailment systems (Roy et al., 2015; Sammons et al., 2010; MacCartney and Manning, 2008; De Marneffe et al., 2008). In medical communications, sole use of verbal specifications (e.g. adjectives and adverbs) has been associated with less precise understanding of frequencies (Nakao and Axelrod, 1983) and probabilities (Timmermans, 1994). A combination of structured data and free text is deemed more suitable for communicating clinical information (Lovis et al., 2000).

Language models have been an integral part of

text prediction systems (Bickel et al., 2005; Wandmacher and Antoine, 2008; Trnka, 2008; Ghosh et al., 2016). Several tasks call for generative language models that have been conditioned on various contexts, e.g. foreign language text for machine translation (Cho et al., 2014), images (Vinyals et al., 2015; Donahue et al., 2015) and videos (Yao et al., 2015) for captioning, etc. Grounded language models represent the relationship between words and the non-linguistic context they refer to. Grounding can help learn better representations for the atoms of language and their interactions. Previous work grounds language on vision (Bruni et al., 2014; Silberer and Lapata, 2014), audio (Kiela and Clark, 2015), video (Fleischman and Roy, 2008) and the olfactory perception (Kiela et al., 2015). Spithourakis et al. (2016) use numerically grounded language models and language models conditioned on a lexicalised knowledge base for the tasks of semantic error detection and correction. We directly use their models to perform word prediction and completion.

3 Methodology

In this section we present a solution to the word prediction and completion tasks (Subsection 3.1). Then, we discuss how language models, which can be grounded on numeric quantities mentioned in the text and/or conditioned on external context can be used in our framework (Subsection 3.2). Finally, we describe our automated evaluation process and various evaluation metrics for the two tasks (Subsection 3.3).

3.1 Word prediction and completion

Let $\{w_1, ..., w_T\}$ denote a document, where w_t is the word at position t. Documents are often associated with external context that can be structured (e.g. a knowledge base) or unstructured (e.g. other documents). Let's consider the case where our context is a knowledge base (KB), that is a set of tuples of the form $< attribute, value >$, where attributes are defined by the KB schema. Different attributes might take values from different domains, e.g. strings, binary values, real numbers etc., and some of the values might be missing.

In the word prediction task, the system presents a ranked list of suggestions for the next word to the user, before the user starts typing. The user can consult this list to explore the vocabulary and guide their decision for the next word to write. The ranking of the items in the list is important, with more strongly endorsed words appearing higher up. Too many displayed options can slow down skilled users (Langlais and Lapalme, 2002), therefore the list should not be too long.

Algorithm 1 Word completion

Input: $\mathcal{V}$ is set of vocabulary words, $scorer$ returns score for word in current position
Output: next word to be written

function COMPLETEWORD($\mathcal{V}$, $scorer$)
  $prefix \leftarrow$ ''
  $lexicon \leftarrow \mathcal{V}$
  loop
    $lexicon \leftarrow$ {tokens in $lexicon$ starting with $prefix$}
    $best \leftarrow \underset{token \in lexicon}{\text{argmax}} \; scorer(token)$
    Display $best$
    $char \leftarrow$ read next char
    if $char = TAB$ **then**
      return best ▷ Auto-complete
    else if $char = WHITESPACE$ **then**
      return $prefix$ ▷ Next word
    else
      $prefix \leftarrow prefix + char$ ▷ Append
    end if
  end loop
end function

Typically, a language model is used to estimate the probability of the next word w_t given the typed word history $w_1, .., w_{t-1}$ and external context. The N-best list of the words with the highest probability is presented as the suggestions.

Word completion is a more interactive task, where the system makes suggestions to complete the current word as the user types each character. Here, the user has a clear intention of typing a specific word and the system should help them achieve this as quickly as possible. A single suggestion is presented and the user can choose to complete the word, typically by typing a special character (e.g. tab).

Word completion is based on interactive prefix matching against a lexicon, as shown in Algo-

rithm 1. The algorithm takes as input the set of known vocabulary words and a scoring function that returns the goodness of a word in the current position and context, which again can be the word probability from a language model. Initialisation sets the prefix to an empty string and the lexicon to the whole vocabulary (lines 2-3). Iteratively, words that do not match with the prefix are removed from the lexicon (line 5), the best word from the lexicon according to the scorer is found and displayed to the user (lines 6-7) and the user can respond with a key (line 8). If the user inputs the special character, the best word is automatically completed (lines 9-10). If the user inputs a whitespace character, the algorithm terminates (11-12). This is the case when no matching word is found in the vocabulary. If any other character is typed, it is appended to the prefix and another iteration begins.

3.2 Neural language models

A language model (LM) estimates the probability of the next token given the previous tokens, i.e. $p(w_t|w_1, ..., w_{t-1})$. Recurrent neural networks (RNNs) have been successfully used for language modelling (Mikolov et al., 2010). Let w_t also denote the one-hot representation of the t-th token, i.e. w_t is a sparse binary vector with a single element set to 1, whose index uniquely identifies the token among a vocabulary of V known words. A neural LM uses a matrix, $E_{in} \in \mathbb{R}^{D \times V}$, to derive word embeddings, $e^w_t = E_{in} w_t$, where D is a latent dimension. A hidden state from the previous time step, h_{t-1}, and the current word embedding, e^w_t, are sequentially fed to an RNN's recurrence function to produce the current hidden state, $h_t \in \mathbb{R}^D$. The conditional probability of the next word is estimated as $\mathrm{softmax}(E_{out} h_t)$, where $E_{out} \in \mathbb{R}^{V \times D}$ is an output embeddings matrix.

We use two extensions to the baseline neural LM, described in Spithourakis et al. (2016). A language model can be *conditioned* on the external context by using an encoder-decoder framework. The encoder builds a representation of the context, h_{KB}, which is then copied to the initial hidden state of the language model (decoder). To build such a representation for our structured context, we can lexicalise the KB by converting its tuples into textual statements of the form "$attribute : value$", which can then be encoded by an RNN. This approach can incorporate KB tuples flexibly, even when values of some attributes are missing.

The document and lexicalised KB will frequently contain numerical tokens, which are typically associated with high out-of-vocabulary rates. To make the LM more sensitive to such numerical information, we can define the inputs of the RNN's recurrence function at each time step as a concatenation of e^w_t and e^n_t, where the latter is a representation of the numeric value of w_t. We set $e^n_t = \mathrm{float}(w_t)$, where $\mathrm{float}(.)$ returns a floating point number from the string of its input or zero, if the conversion fails. When we train such a model, the representations for the words will be associated with the numerical values that appear in their context. Therefore, this model is numerically *grounded*.

3.3 Automated evaluation

We run an automated evaluation for both tasks and all systems by simulating a user who types the text character by character. The character stream comes from a dataset of finalised clinical reports. For the word prediction task, we assume that the word from the dataset is the correct word. For the word completion task, we assume that the user types the special key to autocomplete the word as soon as the correct suggestion becomes available.

In practice, the two tasks can be tackled at the same time, e.g. a list of suggestions based on a language model is shown as the user types and they can choose to complete the prefix with the word on the top of the list. However, we chose to decouple the two functions because of their conceptual differences, which call for different evaluation metrics.

For word prediction, the user has not yet started typing and they might seek guidance in the suggestions of the system for their final decision. A vocabulary exploration system will need to have a high recall. To also capture the effect of the length of the suggestions' list, we will report recall at various ranks (*Recall@k*), where the rank corresponds to the list length. Because our automated evaluation considers a single correct word, Recall@1 is numerically identical to Precision@1. We also report the mean reciprocal rank (*MRR*), which is the multiplicative inverse of the rank of the correct word in the suggestions' list. Finally, per token *perplexity* is

			train	dev	test
#documents			11,158	1,625	3,220
#KB tuples/doc			7.7	7.7	7.7
#tokens/doc		all	204.9	204.4	202.2
		words	95.7%	95.7%	95.7%
		numeric	4.3%	4.3%	4.3%
OOV rate		all	5.0%	5.1%	5.2%
		words	3.4%	3.5%	3.5%
		numeric	40.4%	40.8%	41.8%
#chars/token			4.9	4.9	4.9

Table 1: Statistics for clinical dataset. Counts for non-numeric (*words*) and *numeric* tokens reported as percentage of counts for *all* tokens. Out-of-vocabulary (OOV) rates are for vocabulary of 1000 most frequent words in the train data.

a common evaluation metric for language models.

For word completion, the main goal of the system should be to reduce input time and effort for the intended word that is being typed by the user. *Keystroke savings* (KS) measures the percentage reduction in keys pressed compared to character-by-character text entry. Suggestions that are not taken by the user are a source of unnecessary distractions. We define an *unnecessary distractions* (UD) metric as average number of unaccepted character suggestions that the user has to scan before completing a word.

$$KS = \frac{keys_{unaided} - keys_{with\ prediction}}{keys_{unaided}} \quad (1)$$

$$UD = \frac{count(suggested,\ not\ accepted)}{count(accepted)} \quad (2)$$

Bickel et al. (2005) note that KS corresponds to a recall metric and UD to a precision metric. Thus, we can use the F1 score (harmonic mean of precision and recall) to summarise both metrics.

$$Precision = \frac{count(accepted)}{count(suggested)} \quad (3)$$

$$Recall = \frac{count(accepted)}{count(total\ characters)} \quad (4)$$

4 Data

Our dataset comprises 16,003 anonymised clinical records from the London Chest Hospital. Table 1 summarises descriptive statistics of the dataset.

Each patient record consists of a text report and accompanying structured KB tuples. The latter describe metadata about the patient (age and gender) and results of medical tests (e.g. end diastolic and systolic volumes for the left and right ventricles as measured through magnetic resonance imaging). This information was extracted from the electronic health records held by the hospital and was available to the clinician at the time of the compilation of the report. In total, the KB describes 20 possible attributes. From these, one is categorical (gender) and the rest are numerical (age is integer and test results are real valued). On average, 7.7 tuples are completed per record.

Numeric tokens account for a large part of the vocabulary (>40%) and suffer from high out-of-vocabulary rates (>40%), despite constituting only a small proportion of each sentence (4.3%).

5 Results and discussion

In this section we describe the setup of our experiments (Subsection 5.1) and then present and discuss evaluation results for the word prediction (Subsection 5.2) and word completion (Subsection 5.3) tasks. Finally, we perform a qualitative evaluation (Subsection 5.4).

5.1 Setup

Our *baseline* LM is a single-layer long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) with all latent dimensions (internal matrices, input and output embeddings) set to $D = 50$. We extend this baseline model using the techniques described in Section 3.2 and derive a model conditional on the KB (+*c*), a model that is numerically grounded (+*g*) and a model that is both conditional and grounded (+*c*+*g*). We also experiment with ablations of these models that at test time ignore some source of information. In particular, we run the conditional models without the encoder, which ignores the KB (-*kb*), and the grounded models without the numeric representations, which ignores the magnitudes of the numerical values (-*v*).

	model	PP	MRR	Recall@1	Recall@3	Recall@5	Recall@10
system	baseline	14.96	17.19	8.36	18.38	25.03	36.66
	+c	14.52	54.49	45.27	59.97	65.18	71.18
	+g	9.91	31.91	21.13	35.45	43.66	53.72
	+c+g	**9.39**	**60.71**	**51.76**	**66.36**	**71.28**	**77.10**
ablation	+c -kb	16.64	52.54	43.07	57.89	63.66	70.45
	+g -v	13.16	56.08	46.58	61.96	67.30	73.49
	+c+g -kb	10.82	58.72	49.46	64.31	69.71	75.98
	+c+g -v	11.84	57.31	47.52	63.47	68.92	75.30
	+c+g -kb-v	11.81	56.61	46.68	62.78	68.48	74.87

Table 2: Word-level evaluation results for next word prediction on the test set. Perplexity (PP), mean reciprocal rank (MRR) and Recall at different ranks. Recall@1 is equivalent to Precision@1. Best system values in **bold**.

	model	P	UD	KS(R)	F1
bound	theoretical	100.0	0.00	58.87	74.11
	vocabulary	100.0	0.00	54.48	70.54
system	baseline	13.96	6.17	34.35	19.85
	+c	24.34	3.11	43.17	31.13
	+g	18.60	4.38	39.31	25.25
	+g+c	**26.60**	**2.76**	**44.81**	**33.38**
ablation	+c -kb	24.61	3.06	44.22	31.62
	+g -v	26.74	2.74	45.71	33.74
	+c+g -kb	26.73	2.74	45.72	33.74
	+c+g -v	27.01	2.70	45.86	33.99
	+c+g -kb-v	26.90	2.72	45.79	33.89

Table 3: Character-level evaluation results for word completion on the test set. Unnecessary distractions (UD) is inversely related to precision (P). Keystroke savings (KS) are equivalent with recall (R). Best system values in **bold**.

The vocabulary contains the $V = 1000$ most frequent tokens in the training set. Out-of-vocabulary tokens are substituted with <*num*>, if numeric, and <*unk*>, otherwise. We note that the numerical representations are extracted before any masking. Models are trained to minimise a cross-entropy loss, with 20 epochs of back-propagation and gradient descent with adaptive learing rates (AdaDelta) (Zeiler, 2012) and minibatch size set to 64. Hyperparameters are based on a small search on the development set around values commonly used in the literature.

5.2 Word prediction

We show our evaluation results on the test set for the word prediction task in Table 2. The conditioned model (+c) achieves double the MRR and quadruple the Recall@1 of the baseline model, despite bringing only small improvements in perplexity. The grounded model (+g) achieves a more significant perplexity improvement (33%), but smaller gains for MRR and Recall@1 (85% and 150% improvement, respectively). Contrary to intuition, we observe that a model with higher perplexity performs better in a language modelling task.

The grounded conditional model (+c+g) has the best performance among the systems, with about 5 points additive improvement across all evaluation metrics over the second best. The benefits from conditioning and grounding seem to be orthogonal to one another.

Recall increases with the length of the suggestion list (equivalent to rank). The increase is almost linear for the baseline, but for the grounded conditional it has a decreasing rate. The Recall@5 for the best model is similar to Recall@10 for the second best, thus allowing for halving the suggestions at the same level of recall.

In the test time ablation experiments, all evaluation metrics become slightly worse with the notable exception of the grounded without numerical values (+g-v), for which MRR and recall at all ranks are dramatically increased. Again, we observe that a worse perplexity does not always correlate with decreased performance for the rest of the metrics.

5.3 Word completion

We show our evaluation results on the test set for the word prediction completion in Table 3. In order to give some perspective to the results, we also compute upper bounds originally used to frame

document		system:	baseline	+c	+g	+c+g
left ventricular function			**suggestions**			
analysis results end		1	non	normal	normal	preserved
diastolic volume <num>		2	normal	preserved	dilated	normal
ml end systolic volume	**rank**	3	dilated	non	not	dilated
<num> ml stroke		4	preserved	good	preserved	not
volume <num> ml		5	not	mild	non	with
ejection fraction <num>			**ranks**			
% [...] lv systolic		non-dilated	10	11	8	13
function is moderately		dilated	3	8	2	3
impaired . non dilated	**suggestion**	non	1	3	5	7
atria. non dilated rv [...]		moderately	41	33	37	36
lv is <word> dilated.		mildly	6	6	7	6
[...]		severely	29	23	28	27

Table 4: Word prediction for sample document from the development set. Top-5 suggestion lists for <word> (original document has "non") and ranks for interesting terms from the complete lists of different systems.

		numerical configuration		
		non	mild	severe
<word>	non	***85.83***	**50.45**	26.81
	mildly	11.99	*36.27*	**46.46**
	severely	2.18	13.28	*26.73*

Table 5: Document probabilities for different <word> choices and different numerical configurations. The probabilities are re-normalised over the three displayed choices. Probabilities for highest scoring word in **bold** and for correct word in *italics*.

keystroke savings (Trnka and McCoy, 2008). The *theoretical* bound comes from an ideal system that retrieves the correct word after the user inputs the only the first character. The *vocabulary* bound is similar but only makes any suggestion if the correct word is in the known vocabulary. We extend these bounds to the rest of the evaluation metrics.

The conditioned model (+c) improves the keystroke savings by 25% over the baseline, while halving the unnecessary distractions. The grounded model (+g) achieves smaller improvements over the baseline. The grounded conditional model (+c+g) again has the best performance among the systems. It yields keystroke savings of 44.81%, almost halfway to the theoretical bound, and the lowest number of unnecessary distractions.

For this task, the desired behaviour of a system is to increase the keystroke savings without introducing too many unnecessary distractions (as measured by the number of wrongly suggested characters per word). Since the two quantities represent recall and precision measurements, respectively, a trade-off is expected between them (Bickel et al., 2005). Our extended models manage to improve both quantities without trading one for the other.

The theoretical and vocabulary bounds represent ideal systems that always make correct suggestions (UD=0). This translates into very high precision (100%) and F1 values (>70%) that purely represent upper bounds on these performance metrics. For reference, a system with the same keystroke savings as the theoretical bound (58.87%) and a single unnecessary character per word (UD=1) would achieve precision of 50% and an F1 score of 54.07%.

In the test time ablation experiments, all evaluation metrics have slightly better results than their corresponding system. In fact, some models perform similarly to the best system, if not marginally better.

5.4 Qualitative results

The previous results revealed two unexpected situations. First, we observed that occasionally a model with worse perplexity fares better at word prediction, which is a language modelling task. Second, we observed that occasionally a run time ablation of a conditional or grounded model outperforms its system counterpart. We carried out qualitative experiments in order to investigate these scenarios.

We selected a document from the development

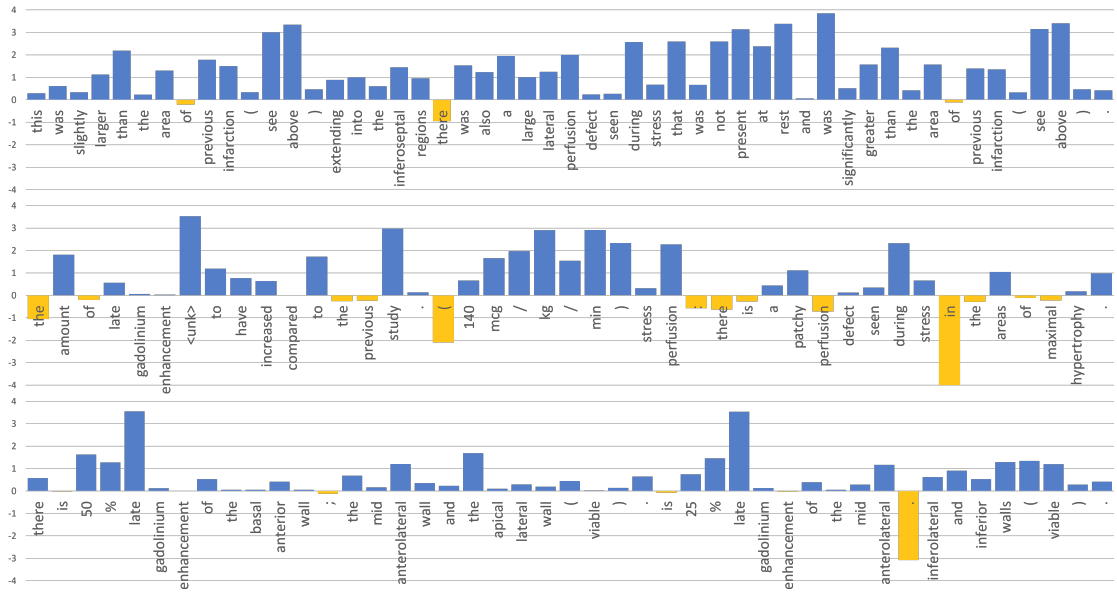

Figure 2: Word likelihood ratios (grounded conditional to baseline) for sample sentences from the development set.

set and identified a word of interest and numeric values that can influence the user's choice for that word. In Table 4, we show the selected document and the 5 top suggestions for the word by different systems. The systems do not have access to tokens from <word> onwards. We also show the ranks for several other semantically relevant choices that appear deeper in the suggestion list. Grounding and conditioning change the order in which the suggestions appear.

We proceeded to substitute the numeric values to more representative configurations that would each favour a particular word choice from the set {"non", "mildly", "severely"}. We found that changing the values does not have a significant effect to the suggestion probabilities and causes no reordering of the items in the lists shown in Table 4. This is in agreement with our previous results for test time ablations and can be attributed to the fact that many more parameters have been used to model words than numerical values. Thus, the systems rely less on numerical information at test time, even though at training time it helps to improve the language models.

Next, for the different numeric configurations we set <word> to each of the three choices and computed the probability of observing the whole document under the grounded model. This is done by multiplying together the probabilities for all individual words. Table 5 shows the resulting document probabilities, re-normalised over the three choices. We observe that the system has a stronger preference to "non", which happens to be the majority class in the training data. In contrast to word probabilities, document probabilities are influenced by the numerical configuration.

The reason for this difference in sensitivities is that the tiny changes in individual word probabilities accumulate multiplicatively to bring on significant changes in the document probability. Additionally, selecting a particular word influences the probabilities of the following words differently, depending on the numerical configuration. This also explains the observed differences between the perplexity of ablated systems, which accumulates small changes over the whole corpus, and the rest of the metrics, which only depend on per word suggestions. Our training objective, cross-entropy, is directly related to perplexity. Through this, numerical values seem to mediate at training time to learn a better language model.

Finally, we directly compare the word probabilities from different systems on several documents from the development set. In Figure 2 we plot the word likelihood ratio of the grounded conditional to baseline language models for three sentences. We

can interpret the values on the vertical axis as how many times the word is more likely under the extended model versus the baseline. The probability of most words was increased, even at longer distances from numbers (first example). This is reflected in the improved perplexity of the language model. Words and contingent spans directly associated with numbers, such as units of measurement and certain symbols, also receive a boost (second example). Finally, the system would often recognise and penalise mistakes because of their unexpectedness (dot instead of a comma in the last example).

6 Conclusion

In this paper we showed how numerically grounded language models conditioned on an external knowledge base can be used in the tasks of word prediction and completion. Our experiments on a clinical dataset showed that the two extensions to standard language models have complimentary benefits. Our best model uses a combination of conditioning and grounding to improve recall from 25.03% to 71.28% for the word prediction task. In the word completion task, it improves keystroke savings from 34.35% to 44.81%, where the upper theoretical bound is 58.78% for this dataset. We found that perplexity does not always correlate with system performance in the two downstream tasks. Our ablation experiments and qualitative investigations showed that at test time numbers have more influence on the document level than on individual word probabilities.

Our approach did not rely on ontologies or fine grained data linkage. Such additional information might lead to further improvements, but would limit the ability of our models to generalise in new settings. While our automated evaluation showed that our extended system achieves notable improvements in keystroke savings, a case study would be required to measure the acceptance of such a system and its impact on clinical documentation processes and patient care. In the past, deployment of text prediction systems in clinical settings has lead to measurable gains in productivity (Hua et al., 2014; Gong et al., 2016).

In the future, we will investigate alternative ways to encode numerical information, in an attempt to improve the utilisation of numerical values at test time. We will also experiment with multitask objectives that consider numerical targets.

Acknowledgments

The authors would like to thank the anonymous reviewers. This research was supported by the Farr Institute of Health Informatics Research and an Allen Distinguished Investigator award.

References

Holger Bast and Ingmar Weber. 2006. Type less, find more: fast autocompletion search with a succinct index. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.

David Beukelman and Pat Mirenda. 2005. *Augmentative and alternative communication*. Brookes.

Steffen Bickel, Peter Haider, and Tobias Scheffer. 2005. Predicting sentences using n-gram language models. In *Proceedings of Human Language Technology and Empirical Methods in Natural Language Processing*.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Mario Cannataro, Orlando Alfieri, and Francesco Fera. 2012. Knowledge-based compilation of magnetic resonance diagnosis reports in neuroradiology. In *25th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE.

Chi-Huang Chen, Sung-Huai Hsieh, Yu-Shuan Su, Kai-Ping Hsu, Hsiu-Hui Lee, and Feipei Lai. 2012. Design and implementation of web-based discharge summary note based on service-oriented architecture. *Journal of medical systems*, 36(1):335–345.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*.

Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding Contradictions in Text. In *Proceedings of ACL*.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Mark D Dunlop and Andrew Crossan. 2000. Predictive text entry methods for mobile phones. *Personal Technologies*, 4(2-3):134–143.

John Eng and Jason M Eisner. 2004. Informatics in radiology (inforad) radiology report entry with automatic phrase completion driven by language modeling. *Radiographics*, 24(5):1493–1501.

Afsaneh Fazly and Graeme Hirst. 2003. Testing the efficacy of part-of-speech information in word completion. In *Proceedings of the EACL 2003 Workshop on Language Modeling for Text Entry Methods*.

Michael Fleischman and Deb Roy. 2008. Grounded Language Modeling for Automatic Speech Recognition of Sports Video. In *Proceedings of ACL*.

George Foster, Philippe Langlais, and Guy Lapalme. 2002. User-friendly text prediction for translators. In *Proceedings of the ACL-02 conference on Empirical methods in natural language*.

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

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv:1602.06291*.

Yang Gong, Lei Hua, and Shen Wang. 2016. Leveraging user's performance in reporting patient safety events by utilizing text prediction in narrative data entry. *Computer methods and programs in biomedicine*, 131:181–189.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

L Hua, S Wang, and Y Gong. 2014. Text prediction on structured data entry in healthcare: A two-group randomized usability study measuring the prediction impact on user performance. *Applied Clinical Informatics*, 5:249–263.

Douwe Kiela and Stephen Clark. 2015. Multi- and Cross-Modal Semantics Beyond Vision: Grounding in Auditory Perception. In *Proceedings of EMNLP*.

Douwe Kiela, Luana Bulat, and Stephen Clark. 2015. Grounding Semantics in Olfactory Perception. In *Proceedings of ACL*.

Philippe Langlais and Guy Lapalme. 2002. Trans type: Development-evaluation cycles to boost translator's productivity. *Machine Translation*, 17(2):77–98.

Ching-Heng Lin, Nai-Yuan Wu, Wei-Shao Lai, and Der-Ming Liou. 2014. Comparison of a semi-automatic annotation tool and a natural language processing application for the generation of clinical statement entries. *Journal of the American Medical Informatics Association*, 22:132–142.

Christian Lovis, Robert H Baud, and Pierre Planche. 2000. Power of expression in the electronic patient record: structured data or narrative text? *International Journal of Medical Informatics*, 58:101–110.

Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*.

Michael A Nakao and Seymour Axelrod. 1983. Numbers are better than words: Verbal specifications of frequency have no place in medicine. *The American Journal of Medicine*, 74(6):1061–1065.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about Quantities in Natural Language. *Transactions of the Association for Computational Linguistics*, 3:1–13.

Mark Sammons, VG Vydiswaran, and Dan Roth. 2010. Ask not what textual entailment can do for you... In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Merlijn Sevenster and Zharko Aleksovski. 2010. Snomed ct saves keystrokes: quantifying semantic autocompletion. In *Proceedings of American Medical Informatics Association (AMIA) Annual Symposium*.

Merlijn Sevenster, Rob van Ommering, and Yuechen Qian. 2012. Algorithmic and user study of an autocompletion algorithm on a large medical vocabulary. *Journal of biomedical informatics*, 45(1):107–119.

Carina Silberer and Mirella Lapata. 2014. Learning Grounded Meaning Representations with Autoencoders. In *Proceedings of ACL*.

Raul Sirel. 2012. Dynamic user interfaces for synchronous encoding and linguistic uniforming of textual clinical data. In *Human Language Technologies–The Baltic Perspective: Proceedings of the 5th International Conference Baltic HLT*.

Georgios P. Spithourakis, Isabelle Augenstein, and Sebastian Riedel. 2016. Numerically grounded language models for semantic error correction. In *Proceedings of EMNLP*.

Daniëlle Timmermans. 1994. The roles of experience and domain of expertise in using numerical and verbal probability terms in medical decisions. *Medical Decision Making*, 14(2):146–156.

Keith Trnka and Kathleen F McCoy. 2008. Evaluating word prediction: framing keystroke savings. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*.

Keith Trnka. 2008. Adaptive language modeling for word prediction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*.

Antal Van Den Bosch and Toine 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*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tonio Wandmacher and Jean-Yves Antoine. 2008. Methods to integrate a language model with semantic information for a word prediction component. In *Proceedings of EMNLP*.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.