Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers

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

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

Abstract

Numeracy is the ability to understand and work with numbers. It is a necessary skill for composing and understanding documents in clinical, scientific, and other technical domains. In this paper, we explore different strategies for modelling numerals with language models, such as memorisation and digit-by-digit composition, and propose a novel neural architecture that uses a continuous probability density function to model numerals from an open vocabulary. Our evaluation on clinical and scientific datasets shows that using hierarchical models to distinguish numerals from words improves a perplexity metric on the subset of numerals by 2 and 4 orders of magnitude, respectively, over non-hierarchical models. A combination of strategies can further improve perplexity. Our continuous probability density function model reduces mean absolute percentage errors by 18% and 54% in comparison to the second best strategy for each dataset, respectively.

1 Introduction

Language models (LMs) are statistical models that assign a probability over sequences of words. Language models can often help with other tasks, such as speech recognition (Mikolov et al., 2010; Prabhavalkar et al., 2017), machine translation (Luong et al., 2015; Gülçehre et al., 2017), text summarisation (Filippova et al., 2015; Gambhir and Gupta, 2017), question answering (Wang et al., 2017), semantic error detection (Rei and Yannakoudakis, 2017; Spithourakis et al., 2016a), and fact checking (Rashkin et al., 2017).

Numeracy and literacy refer to the ability to comprehend, use, and attach meaning to numbers and words, respectively. Language models exhibit literacy by being able to assign higher probabilities to sentences that are both grammatical and realistic, as in this example:

'I eat an apple' (grammatical and realistic)
'An apple eats me' (unrealistic)
'I eats an apple' (ungrammatical)

Likewise, a numerate language model should be able to rank numerical claims based on plausibility:

'John's height is 1.75 metres' (realistic)
'John's height is 999.999 metres' (unrealistic)

Existing approaches to language modelling treat numerals similarly to other words, typically using categorical distributions over a fixed vocabulary.

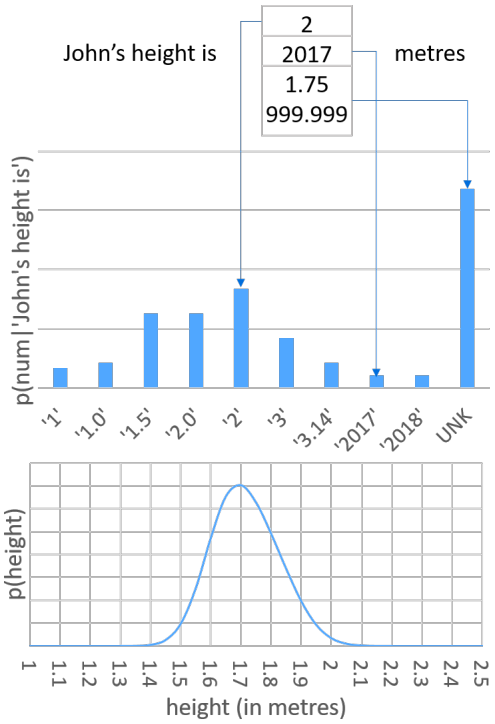

Figure 1: Modelling numerals with a categorical distribution over a fixed vocabulary maps all out-of-vocabulary numerals to the same type, e.g. UNK, and does not reflect the smoothness of the underlying continuous distribution of certain attributes.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 2104–2115
Melbourne, Australia, July 15 - 20, 2018.

However, this maps all unseen numerals to the same unknown type and ignores the smoothness of continuous attributes, as shown in Figure 1. In that respect, existing work on language modelling does not explicitly evaluate or optimise for numeracy. Numerals are often neglected and low-resourced, e.g. they are often masked (Mitchell and Lapata, 2009), and there are only 15,164 (3.79%) numerals among GloVe's 400,000 embeddings pretrained on 6 billion tokens (Pennington et al., 2014). Yet, numbers appear ubiquitously, from children's magazines (Joram et al., 1995) to clinical reports (Bigeard et al., 2015), and grant objectivity to sciences (Porter, 1996).

Previous work finds that numerals have higher out-of-vocabulary rates than other words and proposes solutions for representing unseen numerals as inputs to language models, e.g. using numerical magnitudes as features (Spithourakis et al., 2016b,a). Such work identifies that the perplexity of language models on the subset of numerals can be very high, but does not directly address the issue. This paper focuses on evaluating and improving the ability of language models to predict numerals. The main contributions of this paper are as follows:

1. We explore different strategies for modelling numerals, such as memorisation and digit-by-digit composition, and propose a novel neural architecture based on continuous probability density functions.

2. We propose the use of evaluations that adjust for the high out-of-vocabulary rate of numerals and account for their numerical value (magnitude).

3. We evaluate on a clinical and a scientific corpus and provide a qualitative analysis of learnt representations and model predictions. We find that modelling numerals separately from other words can drastically improve the perplexity of LMs, that different strategies for modelling numerals are suitable for different textual contexts, and that continuous probability density functions can improve the LM's prediction accuracy for numbers.

2 Language Models

Let $s_1,s_2,...,s_L$ denote a document, where s_t is the token at position t. A language model estimates the probability of the next token given previous tokens, i.e. $p(s_t|s_1,...,s_{t-1})$. Neural LMs estimate this probability by feeding embeddings, i.e. vectors that represent each token, into a Recurrent Neural Network (RNN) (Mikolov et al., 2010).

Token Embeddings Tokens are most commonly represented by a D-dimensional dense vector that is unique for each word from a vocabulary $\mathcal{V}$ of known words. This vocabulary includes special symbols (e.g. 'UNK') to handle out-of-vocabulary tokens, such as unseen words or numerals. Let w_s be the one-hot representation of token s, i.e. a sparse binary vector with a single element set to 1 for that token's index in the vocabulary, and $E \in \mathbb{R}^{D \times |\mathcal{V}|}$ be the token embeddings matrix. The token embedding for s is the vector $e_s^{\text{token}} = Ew_s$.

Character-Based Embeddings A representation for a token can be build from its constituent characters (Luong and Manning, 2016; Santos and Zadrozny, 2014). Such a representation takes into account the internal structure of tokens. Let $d_1,d_2,...,d_N$ be the characters of token s. A character-based embedding for s is the final hidden state of a D-dimensional character-level RNN: $e_s^{\text{chars}} = \text{RNN}(d_0,d_1,...d_L)$.

Recurrent and Output Layer The computation of the conditional probability of the next token involves recursively feeding the embedding of the current token e_{s_t} and the previous hidden state h_{t-1} into a D-dimensional token-level RNN to obtain the current hidden state h_t. The output probability is estimated using the softmax function, i.e.

$$p(s_t|h_t) = \text{softmax}(\psi(s_t)) = \frac{1}{Z} e^{\psi(s_t)}$$
$$Z = \sum_{s' \in \mathcal{V}} e^{\psi(s')}, \quad (1)$$

where $\psi(.)$ is a score function.

Training and Evaluation Neural LMs are typically trained to minimise the cross entropy on the training corpus:

$$\mathcal{H}_{train} = -\frac{1}{N} \sum_{s_t \in train} \log p(s_t|s_{<t}) \quad (2)$$

A common performance metric for LMs is per token perplexity (Eq. 3), evaluated on a test corpus. It can also be interpreted as the branching factor: the size of an equally weighted distribution with equivalent uncertainty, i.e. how many sides you need on a fair die to get the same uncertainty as the model distribution.

$$PP_{test} = \exp(\mathcal{H}_{test}) \quad (3)$$

3 Strategies for Modelling Numerals

In this section we describe models with different strategies for generating numerals and propose the

use of number-specific evaluation metrics that adjust for the high out-of-vocabulary rate of numerals and account for numerical values. We draw inspiration from theories of numerical cognition. The triple code theory (Dehaene et al., 2003) postulates that humans process quantities through two exact systems (verbal and visual) and one approximate number system that semantically represents a number on a mental number line. Tzelgov et al. (2015) identify two classes of numbers: i) primitives, which are holistically retrieved from long-term memory; and ii) non-primitives, which are generated online. An in-depth review of numerical and mathematical cognition can be found in Kadosh and Dowker (2015) and Campbell (2005).

3.1 Softmax Model and Variants

This class of models assumes that numerals come from a finite vocabulary that can be memorised and retrieved later. The *softmax* model treats all tokens (words and numerals) alike and directly uses Equation 1 with score function:

$$\psi(s_t) = h_t^T e_{s_t}^{\text{token}} = h_t^T E_{\text{out}} w_{s_t}, \tag{4}$$

where $E_{out} \in \mathbb{R}^{D \times |\mathcal{V}|}$ is an output embeddings matrix. The summation in Equation 1 is over the complete target vocabulary, which requires mapping any out-of-vocabulary tokens to special symbols, e.g. 'UNK$_{\text{word}}$' and 'UNK$_{\text{numeral}}$'.

Softmax with Digit-Based Embeddings The *softmax+rnn* variant considers the internal syntax of a numeral's digits by adjusting the score function:

$$\begin{aligned}\psi(s_t) &= h_t^T e_{s_t}^{\text{token}} + h_t^T e_{s_t}^{\text{chars}} \\ &= h_t^T E_{\text{out}} w_{s_t} + h_t^T E_{\text{out}}^{\text{RNN}} w_{s_t},\end{aligned} \tag{5}$$

where the columns of $E_{\text{out}}^{\text{RNN}}$ are composed of character-based embeddings for in-vocabulary numerals and token embeddings for the remaining vocabulary. The character set comprises digits (0-9), the decimal point, and an end-of-sequence character. The model still requires normalisation over the whole vocabulary, and the special unknown tokens are still needed.

Hierarchical Softmax A hierarchical softmax (Morin and Bengio, 2005a) can help us decouple the modelling of numerals from that of words. The probability of the next token s_t is decomposed to that of its class c_t and the probability of the exact token from within the class:

$$\begin{aligned}p(s_t|h_t) &= \sum_{c_t \in C} p(c_t|h_t) p(s_t|c_t, h_t) \\ p(c_t|h_t) &= \sigma\left(h_t^T b\right)\end{aligned} \tag{6}$$

where the valid token classes are $C = \{\text{word}, \text{numeral}\}$, σ is the sigmoid function and b is a D-dimensional vector. Each of the two branches of $p(s_t|c_t, h_t)$ can now be modelled by independently normalised distributions. The hierarchical variants (*h-softmax* and *h-softmax+rnn*) use two independent softmax distributions for words and numerals. The two branches share no parameters, and thus words and numerals will be embedded into separate spaces.

The hierarchical approach allows us to use any well normalised distribution to model each of its branches. In the next subsections, we examine different strategies for modelling the branch of numerals, i.e. $p(s_t|c_t = \text{numeral}, h_t)$. For simplicity, we will abbreviate this to $p(s)$.

3.2 Digit-RNN Model

Let $d_1, d_2 ... d_N$ be the digits of numeral s. A digit-by-digit composition strategy estimates the probability of the numeral from the probabilities of its digits:

$$p(s) = p(d_1) p(d_2|d_1) ... p(d_N|d_{<N}) \tag{7}$$

The *d-RNN* model feeds the hidden state h_t of the token-level RNN into a character-level RNN (Graves, 2013; Sutskever et al., 2011) to estimate this probability. This strategy can accommodate an open vocabulary, i.e. it eliminates the need for an UNK$_{\text{numeral}}$ symbol, as the probability is normalised one digit at a time over the much smaller vocabulary of digits (digits 0-9, decimal separator, and end-of-sequence).

3.3 Mixture of Gaussians Model

Inspired by the approximate number system and the mental number line (Dehaene et al., 2003), our proposed *MoG* model computes the probability of numerals from a probability density function (pdf) over real numbers, using a mixture of Gaussians for the underlying pdf:

$$\begin{aligned}q(v) &= \sum_{k=1}^{K} \pi_k \mathcal{N}_k(v; \mu_k, \sigma_k^2) \\ \pi_k &= \text{softmax}\left(B^T h_t\right),\end{aligned} \tag{8}$$

where K is the number of components, π_k are mixture weights that depend on hidden state h_t of the token-level RNN, $\mathcal{N}_k$ is the pdf of the normal distribution with mean $\mu_k \in \mathbb{R}$ and variance $\sigma_k^2 \in \mathbb{R}$, and $B \in \mathbb{R}^{D \times K}$ is a matrix.

The difficulty with this approach is that for any continuous random variable, the probability that it equals a specific value is always zero. To resolve this,

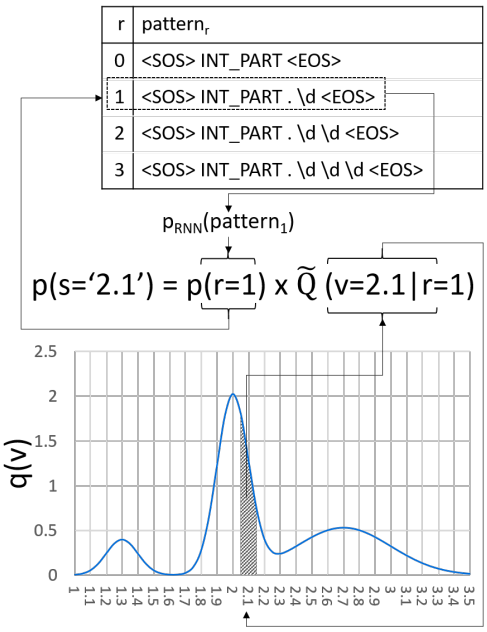

Figure 2: Mixture of Gaussians model. The probability of a numeral is decomposed into the probability of its decimal precision and the probability that an underlying number will produce the numeral when rounded at the given precision.

we consider a probability mass function (pmf) that discretely approximates the pdf:

$$\widetilde{Q}(v|r) = \int_{v-\epsilon_r}^{v+\epsilon_r} q(u)du = F(v+\epsilon_r) - F(v-\epsilon_r), \quad (9)$$

where $F(.)$ is the cumulative density function of $q(.)$, and $\epsilon_r = 0.5 \times 10^{-r}$ is the number's precision. The level of discretisation r, i.e. how many decimal digits to keep, is a random variable in $\mathbb{N}$ with distribution $p(r)$. The mixed joint density is:

$$p(s) = p(v,r) = p(r)\widetilde{Q}(v|r) \quad (10)$$

Figure 2 summarises this strategy, where we model the level of discretisation by converting the numeral into a pattern and use a RNN to estimate the probability of that pattern sequence:

$$p(r) = p(\text{SOS INT\_PART . } \overbrace{\text{\d ... \d}}^{r \text{ decimal digits}} \text{ EOS}) \quad (11)$$

3.4 Combination of Strategies

Different mechanisms might be better for predicting numerals in different contexts. We propose a *combination* model that can select among different strategies for modelling numerals:

$$p(s) = \sum_{\forall m \in M} \alpha_m p(s|m)$$
$$\alpha_m = \text{softmax}\left(A^T h_t\right), \quad (12)$$

where M={h-softmax, d-RNN, MoG}, and $A \in \mathbb{R}^{D \times |M|}$. Since both d-RNN and MoG are open-vocabulary models, the unknown numeral token can now be removed from the vocabulary of h-softmax.

3.5 Evaluating the Numeracy of LMs

Numeracy skills are centred around the understanding of numbers and numerals. A number is a mathematical object with a specific magnitude, whereas a numeral is its symbolic representation, usually in the positional decimal Hindu–Arabic numeral system (McCloskey and Macaruso, 1995). In humans, the link between numerals and their numerical values boosts numerical skills (Griffin et al., 1995).

Perplexity Evaluation Test perplexity evaluated only on numerals will be informative of the symbolic component of numeracy. However, model comparisons based on naive evaluation using Equation 3 might be problematic: perplexity is sensitive to out-of-vocabulary (OOV) rate, which might differ among models, e.g. it is zero for open-vocabulary models. As an extreme example, in a document where all words are out of vocabulary, the best perplexity is achieved by a trivial model that predicts everything as unknown.

Ueberla (1994) proposed Adjusted Perplexity (APP; Eq. 14), also known as unknown-penalised perplexity (Ahn et al., 2016), to cancel the effect of the out-of-vocabulary rate on perplexity. The APP is the perplexity of an adjusted model that uniformly redistributes the probability of each out-of-vocabulary class over all different types in that class:

$$p'(s) = \begin{cases} p(s)\frac{1}{|OOV_c|} & \text{if } s \in OOV_c \\ p(s) & \text{otherwise} \end{cases} \quad (13)$$

where OOV_c is an out-of-vocabulary class (e.g. words and numerals), and $|OOV_c|$ is the cardinality of each OOV set. Equivalently, adjusted perplexity can be calculated as:

$$APP_{test} = \exp\left(\mathcal{H}_{test} + \sum_c \mathcal{H}^c_{\text{adjust}}\right)$$
$$\mathcal{H}^c_{\text{adjust}} = -\sum_t \frac{|s_t \in OOV_c|}{N} \log \frac{1}{|OOV_c|} \quad (14)$$

where N is the total number of tokens in the test set and $|s \in OOV_c|$ is the count of tokens from the test set belonging in each OOV set.

Evaluation on the Number Line While perplexity looks at symbolic performance on numerals, this evaluation focuses on numbers and particularly on their numerical value, which is their most prominent semantic content (Dehaene et al., 2003; Dehaene and Cohen, 1995).

Let v_t be the numerical value of token s_t from the test corpus. Also, let $\hat{v}_t$ be the value of the most probable numeral under the model $s_t = \text{argmax}\,(p(s_t|h_t, c_t = \text{num}))$. Any evaluation metric from the regression literature can be used to measure the models performance. To evaluate on the number line, we can use any evaluation metric from the regression literature. In reverse order of tolerance to extreme errors, some of the most popular are Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Median Absolute Error (MdAE):

$$\begin{aligned} e_i &= v_i - \hat{v}_i \\ RMSE &= \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2} \\ MAE &= \frac{1}{N}\sum_{i=1}^{N} |e_i| \\ MdAE &= \text{median}\{|e_i|\} \end{aligned} \tag{15}$$

The above are sensitive to the scale of the data. If the data contains values from different scales, percentage metrics are often preferred, such as the Mean/Median Absolute Percentage Error (MAPE/MdAPE):

$$\begin{aligned} pe_i &= \frac{v_i - \hat{v}_i}{v_i} \\ MAPE &= \frac{1}{N}\sum_{i=1}^{N} |pe_i| \\ MdAPE &= \text{median}\{|pe_i|\} \end{aligned} \tag{16}$$

4 Data

To evaluate our models, we created two datasets with documents from the clinical and scientific domains, where numbers abound (Bigeard et al., 2015; Porter, 1996). Furthermore, to ensure that the numbers will be informative of some attribute, we only selected texts that reference tables.

Clinical Data Our *clinical* dataset comprises clinical records from the London Chest Hospital. The records where accompanied by tables with 20 numeric attributes (age, heart volumes, etc.) that they partially describe, as well as include numbers not found in the tables. Numeric tokens constitute only a small proportion of each sentence (4.3%), but account for a large part of the unique tokens vocabulary (>40%) and suffer high OOV rates.

Scientific Data Our *scientific* dataset comprises paragraphs from Cornell's ARXIV [1] repository of scientific articles, with more than half a million converted papers in 37 scientific sub-fields. We used the preprocessed ARXMLIV (Stamerjohanns et al., 2010; Stamerjohanns and Kohlhase, 2008) [2] version, where papers have been converted from LATEX into a custom XML format using the LATEXML [3] tool. We then kept all paragraphs with at least one reference to a table and a number.

	Clinical			Scientific		
	Train	**Dev**	**Test**	**Train**	**Dev**	**Test**
#inst	11170	1625	3220	14694	2037	4231
maxLen	667	594	666	2419	1925	1782
avgLen	210.1	209.1	206.9	210.1	215.9	212.1
%word	95.7	95.7	95.7	96.1	96.1	96.0
%nums	4.3	4.3	4.3	3.9	3.9	4.0
min	0.0	0.0	0.0	0.0	0.0	0.0
median	59.5	59.0	60.0	5.0	4.0	4.5
mean	300.6	147.7	464.8	$\sim 10^{21}$	$\sim 10^{7}$	$\sim 10^{7}$
max	$\sim 10^{7}$	$\sim 10^{5}$	$\sim 10^{7}$	$\sim 10^{26}$	$\sim 10^{11}$	$\sim 10^{11}$

Table 1: Statistical description of the clinical and scientific datasets: Number of instances (i.e. paragraphs), maximum and average lengths, proportions of words and numerals, descriptive statistics of numbers.

For both datasets, we lowercase tokens and normalise numerals by omitting the thousands separator ("2,000" becomes "2000") and leading zeros ("007" becomes "7"). Special mathematical symbols are tokenised separately, e.g. negation ("-1" as "-", "1"), fractions ("3/4" as "3", "/", "4"), etc. For this reason, all numbers were non-negative. Table 1 shows descriptive statistics for both datasets.

5 Experimental Results and Discussion

We set the vocabularies to the 1,000 and 5,000 most frequent token types for the clinical and scientific datasets, respectively. We use gated token-character embeddings (Miyamoto and Cho, 2016) for the input of numerals and token embeddings for the input and output of words, since the scope of our paper is numeracy. We set the models' hidden dimensions to $D = 50$ and initialise all token embeddings to pretrained GloVe (Pennington et al., 2014). All our

[1] ARXIV.ORG. Cornell University Library at http://arxiv.org/, visited December 2016

[2] ARXMLIV. Project home page at http://arxmliv.kwarc.info/, visited December 2016

[3] LATEXML. http://dlmf.nist.gov, visited December 2016

	Clinical						Scientific					
	words		numerals		total		words		numerals		total	
Model	**PP**	**APP**	**PP**	**APP**	**PP**	**APP**	**PP**	**APP**	**PP**	**APP**	**PP**	**APP**
softmax	4.08	5.99	12.04	58443.72	4.28	8.91	33.96	51.83	127.12	3505856.25	35.79	80.62
softmax+rnn	4.03	5.91	**11.57**	56164.81	4.21	8.77	**33.54**	51.20	**119.68**	3300688.50	**35.28**	79.47
h-softmax	**4.00**	4.96	11.78	495.95	**4.19**	6.05	34.73	49.81	122.67	550.98	36.51	54.80
h-softmax+rnn	4.03	4.99	11.65	490.14	4.22	6.09	34.04	48.83	120.83	542.70	35.80	53.73
d-RNN	3.99	**4.95**	263.22	263.22	4.79	5.88	34.08	48.89	519.80	**519.80**	37.98	53.70
MoG	4.03	4.99	226.46	226.46	4.79	5.88	34.14	48.97	683.16	683.16	38.45	54.37
combination	4.01	4.96	197.59	**197.59**	4.74	**5.82**	33.64	**48.25**	520.95	520.95	37.50	**53.03**

Table 2: Test set perplexities for the clinical and scientific data. Adjusted perplexities (APP) are directly comparable across all data and models, but perplexities (PP) are sensitive to the varying out-of-vocabulary rates.

	Clinical					Scientific		
Model	**RMSE**	**MAE**	**MdAE**	**MAPE%**	**MdAPE%**	**MdAE**	**MAPE%**	**MdAPE%**
mean	1043.68	294.95	245.59	2353.11	409.47	$\sim 10^{20}$	$\sim 10^{23}$	$\sim 10^{22}$
median	1036.18	120.24	34.52	425.81	52.05	4.20	8039.15	98.65
softmax	997.84	80.29	12.70	621.78	22.41	3.00	1947.44	80.62
softmax+rnn	991.38	74.44	13.00	503.57	23.91	3.50	15208.37	80.00
h-softmax	1095.01	167.19	14.00	746.50	25.00	3.00	1652.21	80.00
h-softmax+rnn	1001.04	83.19	12.30	491.85	23.44	3.00	2703.49	80.00
d-RNN	1009.34	70.21	9.00	513.81	17.90	3.00	1287.27	**52.45**
MoG	998.78	**57.11**	**6.92**	**348.10**	**13.64**	**2.10**	**590.42**	90.00
combination	**989.84**	69.47	9.00	552.06	17.86	3.00	2332.50	88.89

Table 3: Test set regression evaluation for the clinical and scientific data. Mean absolute percentage error (MAPE) is scale independent and allows for comparison across data, whereas root mean square and mean absolute errors (RMSE, MAE) are scale dependent. Medians (MdAE, MdAPE) are informative of the distribution of errors.

RNNs are LSTMs (Hochreiter and Schmidhuber, 1997) with the biases of LSTM forget gate were initialised to 1.0 (Józefowicz et al., 2015). We train using mini-batch gradient decent with the Adam optimiser (Kingma and Ba, 2014) and regularise with early stopping and 0.1 dropout rate (Srivastava, 2013) in the input and output of the token-based RNN.

For the mixture of Gaussians, we select the mean and variances to summarise the data at different granularities by fitting 7 separate mixture of Gaussian models on all numbers, each with twice as many components as the previous, for a total of $2^{7+1} - 1 = 256$ components. These models are initialised at percentile points from the data and trained with the expectation-minimisation algorithm. The means and variances are then fixed and not updated when we train the language model.

5.1 Quantitative Results

Perplexities Table 2 shows perplexities evaluated on the subsets of words, numerals and all tokens of the test data. Overall, all models performed better on the clinical than on the scientific data. On words, all models achieve similar perplexities in each dataset.

On numerals, softmax variants perform much better than other models in PP, which is an artefact of the high OOV-rate of numerals. APP is significantly worse, especially for non-hierarchical variants, which perform about 2 and 4 orders of magnitude worse than hierarchical ones.

For open-vocabulary models, i.e. d-RNN, MoG, and combination, PP is equivalent to APP. On numerals, d-RNN performed better than softmax variants in both datasets. The MoG model performed twice as well as softmax variants on the clinical dataset, but had the third worse performance in the scientific dataset. The combination model had the best overall APP results for both datasets.

Evaluations on the Number Line To factor out model specific decoding processes for finding the best next numeral, we use our models to rank a set

of candidate numerals: we compose the union of in-vocabulary numbers and 100 percentile points from the training set, and we convert numbers into numerals by considering all formats up to n decimal points. We select n to represent 90% of numerals seen at training, which yields $n=3$ and $n=4$ for the clinical and scientific data, respectively.

Table 3 shows evaluation results, where we also include two naive baselines of constant predictions: with the mean and median of the training data. For both datasets, RMSE and MAE were too sensitive to extreme errors to allow drawing safe conclusions, particularly for the scientific dataset, where both metrics were in the order of 10^9. MdAE can be of some use, as 50% of the errors are absolutely smaller than that.

Along percentage metrics, MoG achieved the best MAPE in both datasets (18% and 54% better that the second best) and was the only model to perform better than the median baseline for the clinical data. However, it had the worst MdAPE, which means that MoG mainly reduced larger percentage errors. The d-RNN model came third and second in the clinical and scientific datasets, respectively. In the latter it achieved the best MdAPE, i.e. it was effective at reducing errors for 50% of the numbers. The combination model did not perform better than its constituents. This is possibly because MoG is the only strategy that takes into account the numerical magnitudes of the numerals.

5.2 Learnt Representations

Softmax versus Hierarchical Softmax Figure 3 visualises the cosine similarities of the output token embeddings of numerals for the softmax and h-softmax models. Simple softmax enforced high similarities among all numerals and the unknown numeral token, so as to make them more dissimilar to words, since the model embeds both in the same space. This is not the case for h-softmax that uses two different spaces: similarities are concentrated along the diagonal and fan out as the magnitude grows, with the exception of numbers with special meaning, e.g. years and percentile points.

Digit embeddings Figure 4 shows the cosine similarities between the digits of the d-RNN output mode. We observe that each primitive digit is mostly similar to its previous and next digit. Similar behaviour was found for all digit embeddings of all models.

5.3 Predictions from the Models

Next Numeral Figure 5 shows the probabilities of different numerals under each model for two

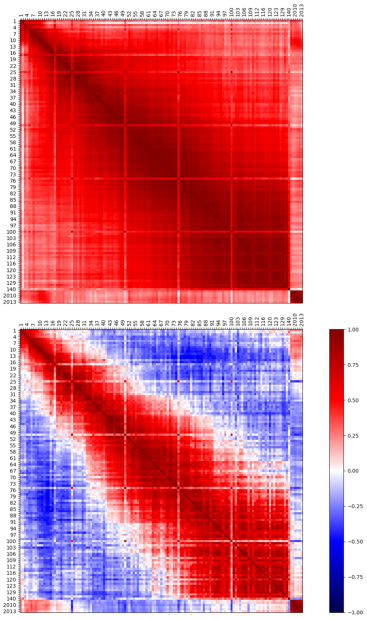

Figure 3: Numeral embeddings for the softmax (top) and h-softmax (bottom) models on the clinical data. Numerals are sorted by value.

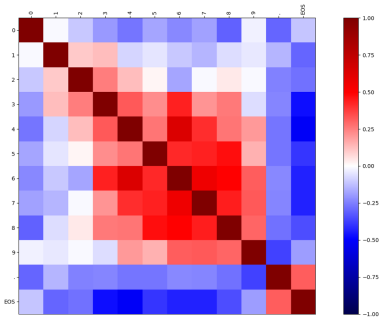

Figure 4: Cosine similarities for d-RNN's output digit embeddings trained on the scientific data.

examples from the clinical development set. Numerals are grouped by number of decimal points. The h-softmax model's probabilities are spiked, d-RNNs are saw-tooth like and MoG's are smooth, with the occasional spike, whenever a narrow component allows for it. Probabilities rapidly decrease for more decimal digits, which is reminiscent of the theoretical expectation that the probability of en exact value for a continuous variable is zero.

Selection of Strategy in Combination Model Table 4 shows development set examples with high selection probabilities for each strategy of the combination model, along with numerals with the highest average selection per mode. The h-softmax model is responsible for mostly integers with special functions,

	Clinical	Scientific
h-softmax	**Examples**: *"late enhancement (> **75** %)", "late gadolinium enhancement (< **25** %)", "infarction (**2** out of **17** segments)", "infarct with **4** out of **17** segments nonviable", "adenosine stress perfusion @ **140** mcg", "stress perfusion (adenosine **140** mcg"* **Numerals**: 50, 17, 100, 75, 25, 1, 140, 2012, 2010, 2011, 8, 5, 2009, 2013, 7, 6, 2, 3, 2008, 4...	**Examples**: *"sharp et al . **2004**", "li et al . **2003**", "3.5 × **10**^4", "0.3 × **10**^16"* **Numerals**: 1992, 2001, 1995, 2003, 2009, 1993, 2010, 1994, 1998, 2002, 2006, 1997, 2005, 1990, 10, 2008, 2007, 2004, 1983, 1991...
d-RNN	**Examples**: *"aortic root is dilated (measured **37** x **37** mm", "ascending aorta is not dilated (**32** x **31** mm"* **Numerals**: 42, 33, 31, 43, 44, 21, 38, 36, 46, 37, 32, 39, 26, 28, 23, 29, 45, 40, 49, 94...	**Examples**: *"ngc **6334** stars", "ngc **2366** shows a wealth of small structures"* **Numerals**: 294, 4000, 238, 6334, 2363, 1275, 2366, 602, 375, 1068, 211, 6.4, 8.7, 600, 96, 0.65, 700, 1.17, 4861, 270...
MoG	**Examples**: *"stroke volume **46.1** ml", "stroke volume **65.6** ml", "stroke volume **74.5** ml", "end diastolic volume **82.6** ml", "end diastolic volume **99.09** ml", "end diastolic volume **138.47** ml"* **Numerals**: 74.5, 69.3, 95.9, 96.5, 72.5, 68.6, 82.1, 63.7, 78.6, 69.6, 69.5, 82.2, 68.3, 73.2, 63.2, 82.6, 77.7, 80.7, 70.7, 70.4...	**Examples**: *"hip **12961** and gl **676** a are orbited by giant planets," "velocities of gl **676**", "velocities of hip **12961"*** **Numerals**: 12961, 766, 7409, 4663, 44.3, 1819, 676, 1070, 5063, 323, 264, 163296, 2030, 77, 1.15, 196, 0.17, 148937, 0.43, 209458...

Table 4: Examples of numerals with highest probability in each strategy of the combination model.

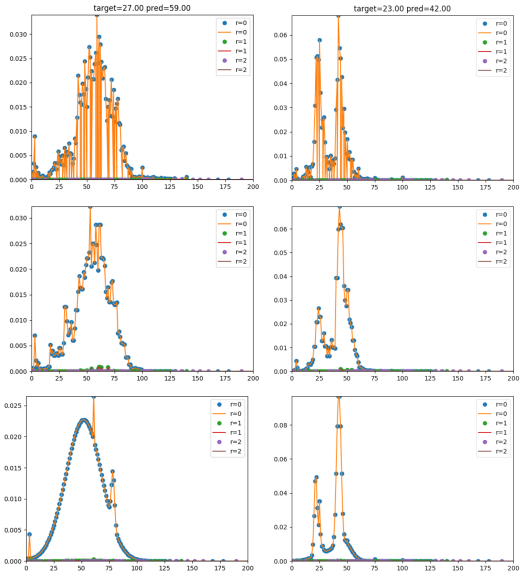

Figure 5: Example model predictions for the h-softmax (top), d-RNN (middle) and MoG (bottom) models. Examples from the clinical development set.

e.g. years, typical drug dosages, percentile points, etc. In the clinical data, d-RNN picks up two-digit integers (mostly dimensions) and MoG is activated for continuous attributes, which are mostly out of vocabulary. In the scientific data, d-RNN and MoG

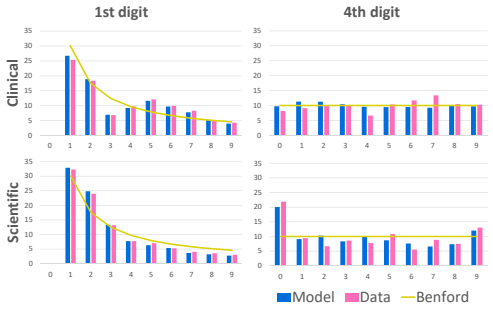

Figure 6: Distributions of significant digits from d-RNN model, data, and theoretical expectation (Benford's law).

showed affinity to different indices from catalogues of astronomical objects: d-RNN mainly to NGC (Dreyer, 1888) and MoG to various other indices, such as GL (Gliese, 1988) and HIP (Perryman et al., 1997). In this case, MoG was wrongly selected for numerals with a labelling function, which also highlights a limitation of evaluating on the number line, when a numeral is not used to represent its magnitude.

Significant Digits Figure 5 shows the distributions of the most significant digits under the d-RNN model

and from data counts. The theoretical estimate has been overlayed, according to Benford's law (Benford, 1938), also called the first-digit law, which applies to many real-life numerals. The law predicts that the first digit is 1 with higher probability (about 30%) than 9 ($< 5\%$) and weakens towards uniformity at higher digits. Model probabilities closely follow estimates from the data. Violations from Benford's law can be due to rounding (Beer, 2009) and can be used as evidence for fraud detection (Lu et al., 2006).

6 Related Work

Numerical quantities have been recognised as important for textual entailment (Lev et al., 2004; Dagan et al., 2013). Roy et al. (2015) proposed a quantity entailment sub-task that focused on whether a given quantity can be inferred from a given text and, if so, what its value should be. A common framework for acquiring common sense about numerical attributes of objects has been to collect a corpus of numerical values in pre-specified templates and then model attributes as a normal distribution (Aramaki et al., 2007; Davidov and Rappoport, 2010; Iftene and Moruz, 2010; Narisawa et al., 2013; de Marneffe et al., 2010). Our model embeds these approaches into a LM that has a sense for numbers.

Other tasks that deal with numerals are numerical information extraction and solving mathematical problems. Numerical relations have at least one argument that is a number and the aim of the task is to extract all such relations from a corpus, which can range from identifying a few numerical attributes (Nguyen and Moschitti, 2011; Intxaurrondo et al., 2015) to generic numerical relation extraction (Hoffmann et al., 2010; Madaan et al., 2016). Our model does not extract values, but rather produces an probabilistic estimate.

Much work has been done in solving arithmetic (Mitra and Baral, 2016; Hosseini et al., 2014; Roy and Roth, 2016), geometric (Seo et al., 2015), and algebraic problems (Zhou et al., 2015; Koncel-Kedziorski et al., 2015; Upadhyay et al., 2016; Upadhyay and Chang, 2016; Shi et al., 2015; Kushman et al., 2014) expressed in natural language. Such models often use mathematical background knowledge, such as linear system solvers. The output of our model is not based on such algorithmic operations, but could be extended to do so in future work.

In language modelling, generating rare or unknown words has been a challenge, similar to our unknown numeral problem. Gulcehre et al. (2016) and Gu et al. (2016) adopted pointer networks (Vinyals et al., 2015) to copy unknown words from the source in translation and summarisation tasks. Merity et al. (2016) and Lebret et al. (2016) have models that copy from context sentences and from Wikipedia's infoboxes, respectively. Ahn et al. (2016) proposed a LM that retrieves unknown words from facts in a knowledge graph. They draw attention to the inappropriateness of perplexity when OOV-rates are high and instead propose an adjusted perplexity metric that is equivalent to APP. Other methods aim at speeding up LMs to allow for larger vocabularies (Chen et al., 2015), such as hierarchical softmax (Morin and Bengio, 2005b), target sampling (Jean et al., 2014), etc., but still suffer from the unknown word problem. Finally, the problem is resolved when predicting one character at a time, as done by the character-level RNN (Graves, 2013; Sutskever et al., 2011) used in our d-RNN model.

7 Conclusion

In this paper, we investigated several strategies for LMs to model numerals and proposed a novel open-vocabulary generative model based on a continuous probability density function. We provided the first thorough evaluation of LMs on numerals on two corpora, taking into account their high out-of-vocabulary rate and numerical value (magnitude). We found that modelling numerals separately from other words through a hierarchical softmax can substantially improve the perplexity of LMs, that different strategies are suitable for different contexts, and that a combination of these strategies can help improve the perplexity further. Finally, we found that using a continuous probability density function can improve prediction accuracy of LMs for numbers by substantially reducing the mean absolute percentage metric.

Our approaches in modelling and evaluation can be used in future work in tasks such as approximate information extraction, knowledge base completion, numerical fact checking, numerical question answering, and fraud detection. Our code and data are available at: `https://github.com/uclmr/numerate-language-models`.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments and also Steffen Petersen for providing the clinical dataset and advising us on the clinical aspects of this work. This research was supported by the Farr Institute of Health Informatics Research and an Allen Distinguished Investigator award.

References

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *arXiv preprint arXiv:1608.00318* .

Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. 2007. Uth: Svm-based semantic relation classification using physical sizes. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, pages 464–467.

TW Beer. 2009. Terminal digit preference: beware of benford's law. *Journal of clinical pathology* 62(2):192–192.

Frank Benford. 1938. The law of anomalous numbers. *Proceedings of the American philosophical society* pages 551–572.

Elise Bigeard, Vianney Jouhet, Fleur Mougin, Frantz Thiessard, and Natalia Grabar. 2015. Automatic extraction of numerical values from unstructured data in ehrs. In *MIE*. pages 50–54.

Jamie ID Campbell. 2005. *Handbook of mathematical cognition*. Psychology Press.

Welin Chen, David Grangier, and Michael Auli. 2015. Strategies for training large vocabulary neural language models. *arXiv preprint arXiv:1512.04906* .

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies* 6(4):1–220.

Dmitry Davidov and Ari Rappoport. 2010. Extraction and approximation of numerical attributes from the web. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1308–1317.

Marie-Catherine de Marneffe, Christopher D Manning, and Christopher Potts. 2010. Was it good? it was provocative. learning the meaning of scalar adjectives. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 167–176.

Stanislas Dehaene and Laurent Cohen. 1995. Towards an anatomical and functional model of number processing. *Mathematical cognition* 1(1):83–120.

Stanislas Dehaene, Manuela Piazza, Philippe Pinel, and Laurent Cohen. 2003. Three parietal circuits for number processing. *Cognitive neuropsychology* 20(3-6):487–506.

John Louis Emil Dreyer. 1888. A new general catalogue of nebulæ and clusters of stars, being the catalogue of the late sir john fw herschel, bart, revised, corrected, and enlarged. *Memoirs of the Royal Astronomical Society* 49:1.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. pages 360–368.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artif. Intell. Rev.* 47(1):1–66.

Wilhelm Gliese. 1988. The third catalogue of nearby stars. *Stand. Star Newsl. 13, 13* 13.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

Sharon Griffin, Robbie Case, and Allesandra Capodilupo. 1995. Teaching for understanding: The importance of the central conceptual structures in the elementary mathematics curriculum. .

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393* .

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *arXiv preprint arXiv:1603.08148* .

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language* 45:137–148.

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

Raphael Hoffmann, Congle Zhang, and Daniel S Weld. 2010. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 286–295.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 523–533.

Adrian Iftene and Mihai-Alex Moruz. 2010. Uaic participation at rte-6 .

Ander Intxaurrondo, Eneko Agirre, Oier Lopez De Lacalle, and Mihai Surdeanu. 2015. Diamonds in the rough: Event extraction from imperfect microblog data. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 641–650.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007* .

Elana Joram, Lauren B Resnick, and Anthony J Gabriele. 1995. Numeracy as cultural practice: An examination of numbers in magazines for children, teenagers, and adults. *Journal for Research in Mathematics Education* pages 346–361.

Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*. pages 2342–2350.

Roi Cohen Kadosh and Ann Dowker. 2015. *The Oxford handbook of numerical cognition*. Oxford Library of Psychology.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics* 3:585–597.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 271–281.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771* .

Iddo Lev, Bill MacCartney, Christopher D Manning, and Roger Levy. 2004. Solving logic puzzles: From robust processing to precise semantics. In *Proceedings of the 2nd Workshop on Text Meaning and Interpretation*. Association for Computational Linguistics, pages 9–16.

Fletcher Lu, J. Efrim Boritz, and H. Dominic Covvey. 2006. Adaptive fraud detection using benford's law. In *Advances in Artificial Intelligence, 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006*. pages 347–358.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1054–1063.

Thang Luong, Michael Kayser, and Christopher D. Manning. 2015. Deep neural language models for machine translation. In *Proceedings of the 19th Conference on Computational Natural Language Learning, CoNLL 2015*. pages 305–309.

Aman Madaan, Ashish Mittal, Ganesh Ramakrishnan, Sunita Sarawagi, et al. 2016. Numerical relation extraction with minimal supervision. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Michael McCloskey and Paul Macaruso. 1995. Representing and using numerical information. *American Psychologist* 50(5):351.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843* .

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*. pages 1045–1048.

Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, pages 430–439.

Arindam Mitra and Chitta Baral. 2016. Learning to use formulas to solve simple arithmetic problems. In *ACL*.

Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. *arXiv preprint arXiv:1606.01700* .

Frederic Morin and Yoshua Bengio. 2005a. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005*.

Frederic Morin and Yoshua Bengio. 2005b. Hierarchical probabilistic neural network language model. In *Aistats*. Citeseer, volume 5, pages 246–252.

Katsuma Narisawa, Yotaro Watanabe, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui. 2013. Is a 204 cm man tall or small? acquisition of numerical common sense from the web. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 382–391.

Truc-Vien T Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pages 277–282.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Michael AC Perryman, L Lindegren, J Kovalevsky, E Hoeg, U Bastian, PL Bernacca, M Crézé, F Donati, M Grenon, M Grewing, et al. 1997. The hipparcos catalogue. *Astronomy and Astrophysics* 323:L49–L52.

Theodore M Porter. 1996. *Trust in numbers: The pursuit of objectivity in science and public life*. Princeton University Press.

Rohit Prabhavalkar, Kanishka Rao, Tara N. Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. 2017. A comparison of sequence-to-sequence models for speech recognition. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*. pages 939–943.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*. pages 2931–2937.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@EMNLP 2017*. pages 33–43.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413* .

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

Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1818–1826.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1466–1476.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*.

Georgios P. Spithourakis, Isabelle Augenstein, and Sebastian Riedel. 2016a. Numerically grounded language models for semantic error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*. pages 987–992.

Georgios P Spithourakis, Steffen E Petersen, and Sebastian Riedel. 2016b. Clinical text prediction with numerically grounded conditional language models. *EMNLP 2016* page 6.

Nitish Srivastava. 2013. Improving neural networks with dropout. *University of Toronto* 182.

Heinrich Stamerjohanns and Michael Kohlhase. 2008. Transforming the arχiv to xml. In *International Conference on Intelligent Computer Mathematics*. Springer, pages 574–582.

Heinrich Stamerjohanns, Michael Kohlhase, Deyan Ginev, Catalin David, and Bruce Miller. 2010. Transforming large collections of scientific publications to xml. *Mathematics in Computer Science* 3(3):299–307.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. pages 1017–1024.

Joseph Tzelgov, Dana Ganor-Stern, Arava Y Kallai, and Michal Pinhas. 2015. Primitives and non-primitives of numerical representations. *Oxford library of psychology. The Oxford handbook of numerical cognition* pages 45–66.

Joerg Ueberla. 1994. Analysing a simple language model· some general conclusions for language models for speech recognition. *Computer Speech & Language* 8(2):153–176.

Shyam Upadhyay and Ming-Wei Chang. 2016. Annotating derivations: A new evaluation strategy and dataset for algebra word problems. *arXiv preprint arXiv:1609.07197* .

Shyam Upadhyay, Ming-Wei Chang, Kai-Wei Chang, and Wen-tau Yih. 2016. Learning from explicit and implicit supervision jointly for algebra word problems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 297–306.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*. pages 2692–2700.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *CoRR* abs/1706.01450.

Lipu Zhou, Shuaixiang Dai, and Liwei Chen. 2015. Learn to solve algebra word problems using quadratic programming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (Lisbon, Portugal*. pages 817–822.