Whose Line Is It? – Quote Attribution through Recurrent Neural Networks

Edward Schmerling
schmrlng@stanford.edu

Abstract

This paper presents a recurrent neural network framework for the problem of attributing spoken lines to characters in a screenplay or novel. We study these quotes as a sequence in the absence of additional context, e.g. descriptions of scenes or actions, from the text surrounding them. Instead, attributions may only be made on the basis of learned expectations for how each character speaks, as well as an understanding of how they converse with each other. We use gated-feedback recurrent neural networks, trained in a supervised fashion, for modeling both of these aspects. We evaluate the prediction model on episodes of the television show *Futurama* and demonstrate improvement over simpler neural network constructions.

1 Introduction

Conversation is a fundamental means for human interaction. Following the back-and-forth thread of a conversation, as an interactive participant, is essential for constructing mutual understanding, but beyond this core competence we are generally able to drop in on a discussion already underway and similarly grasp its flow. Furthermore, after prolonged observation of the same parties conversing, we start to infer mental models for the speakers and develop expectations for what each speaker is more likely to say. These speaker models transcend the audible aspects of spoken conversation, and are founded instead upon the underlying language.

In this work we consider the task of *quote attribution* (alternately, *speaker identification*) in the context of dialogue extracted from literary novels and television or movie screenplays. That is, given a sequence of unlabeled dialogue text, the aim is to classify each line according to its speaker. Performing this task well requires both an understanding of the temporal flow and the semantic substance of the dialogue lines. For example, speakers in a scene are likely to alternate while producing utterances in keeping with their personalities endowed by their authors.

This problem of purely language-based speaker identification, without the aid of auditory signals or visual cues, has been previously studied in [2, 4, 5]. In addition to the bare quotations spoken by each of the characters, these works also consider the addition of context features (e.g., "she said, ..." or "he said, ..." which imply speaker gender). We note that the approach outlined in this paper does not make use of such features, but attempts instead to classify only on the basis of the language of the current and previous quotations. We call this modified setup the *dialogue-only quote attribution* problem in this work. Both [4] and [5] take into account past conversational flow when attributing quotes. In [4], it is assumed that all previous lines are classified correctly according to their speaker (i.e., through an oracle) as features present when classifying the current line. He et al. [5] forgo this arguably unrealistic assumption by employing a manually constructed model of speaker alternation patterns. Chaganty and Muzny [2] apply a neural network-based model for the quote attribution problem but attends only to the semantic substance of dialogue; their model is not capable of capturing the notion of an ongoing conversation.

In this paper we address the problem of speaker identification through the lens of recurrent neural networks (RNNs). RNN-based approaches to modeling language have recently achieved state-of-the-art performance [8] particularly owing to their ability to capture context (or in the case of streaming dialogue, "history") of arbitrary length. RNNs thus seem a natural choice for learning the likelihood of a quote belonging to the lexicon of a particular character, as well as how to modify that likelihood conditioned on the prior sequence of conversation. RNNs capture context by maintaining a hidden state at each timestep, in addition to the output used for prediction, that emulates a condensed summary of the preceding timesteps. There are a number of hidden unit constructions of varying depth and complexity commonly used for updating this state; in this work we focus on gated-feedback RNNs (GF-RNNs) [3]. The primary contribution of this work is to evaluate the merits of using bidirectional GF-RNNs, at both the word level and line level, for dialogue-only quote attribution over simpler recurrent and non-recurrent neural network models.

2 Problem Statement

We consider a corpus $\mathcal{C} = \{\mathcal{E}_1, \ldots, \mathcal{E}_K\}$ consisting of dialogue extracted from a set of television episodes (or chapters, in the case of a literary novel). Each episode $\mathcal{E}_k = \{\ell_1^{(k)}, \ldots, \ell_{N_k}^{(k)}\}$ is a sequence of dialogue lines, and each line $\ell_l^{(k)} = \{w_1^{(k,l)}, \ldots, w_{M_{k,l}}^{(k,l)}\}$ is a sequence of word tokens, possibly spanning multiple sentences. Each line $\ell_l^{(k)}$ is associated with a label $s_l^{(k)}$ denoting its speaker. Given a corpus of training episodes from the same television show, we consider the dialogue-only quote attribution problem of assigning speaker labels to lines from a previously unseen episode $\tilde{\mathcal{E}}$ stripped of speaker tags. We evaluate the performance of a prediction model using a weighted average cross-entropy loss; we weight the loss to improve classification accuracy for less common speakers. To be precise, given one-hot ground truth speaker label vectors $\mathbf{y}^{(l)}$ and corresponding predicted probability vector $\hat{\mathbf{y}}^{(l)}$, we wish to minimize

$$\begin{aligned} J &= \frac{1}{N_{\text{lines}}} \sum_{l=1}^{N_{\text{lines}}} \mathbf{w}^T \mathbf{y}^{(l)} CE(\mathbf{y}^{(l)}, \hat{\mathbf{y}}^{(l)}) \\ &= \frac{1}{N_{\text{lines}}} \sum_{l=1}^{N_{\text{lines}}} \mathbf{w}^T \mathbf{y}^{(l)} \left(\sum_{k=1}^{N_{\text{speakers}}} y_k^{(l)} \log(\hat{y}_k^{(l)}) \right) \end{aligned} \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{N_{\text{speakers}}}$ is a weight vector with higher values for characters that speak less. In the numerical experiments Section 4, we also present F_1 scores (for discrete predictions) for each character as a way to understand classification accuracy by class.

2.1 Data

The dataset for this work consists of all episode screenplays from seasons 1–5 of *Futurama* scraped from `http://www.imsdb.com/`, the Internet Movie Script Database (IMSDb) For the purposes of training and evaluation, each line is labeled by the episode number as well as its true speaker label. No immediate context beyond utterance ordering is taken from the screenplays or novel. We split the 72 *Futurama* episodes randomly as 54/8/8 training/validation/test; the specific split may be obtained at `https://github.com/schmrlng/RNNQuoteAttribution`.

3 Technical Approach

3.1 Preprocessing and Word Vectors

We use the GloVe word vectors [9] trained on a 6 billion token Gigaword5 + Wikipedia2014 corpus as the initial translation layer in all of the models considered in this paper. We tokenize and lowercase all quotes in our corpus using the Stanford tokenizer [7] in order to be consistent with the GloVe data. As the *Futurama* dataset is relatively small, consisting of 175K tokens spread over 15061 quotations, we do not retrain the word vectors when training the models described below. Given input tokens $\{w_t^{(l)}\}$ from dialogue lines (let us assume we are working within a single episode and omit the superscript k), let $\{\mathbf{x}_t^{(l)}\}$ be one-hot row vectors into the GloVe embedding matrix $\mathbf{L} \in \mathbb{R}^{|V|\times d}$. Then this first layer may be summarized by the equation

$$\mathbf{w}_t^{(l)} = \mathbf{x}_t^{(l)}\mathbf{L}. \tag{2}$$

3.2 Neural Network Model

As described in the introduction, there are two main timescales to consider in quote attribution: (1) the sequence of words within a particular dialogue line, and (2) the sequence of lines within an episode. We model each of these timescales with a neural network which we stack to produce the full prediction model. Layer 1 (which may actually consist of multiple sub-layers) maps an input sequence of word vectors $\{\mathbf{w}_1^{(l)}, \ldots, \mathbf{w}_{M_l}^{(l)}\}$ to a quote vector $\mathbf{q}_l$. Loosely speaking, the intention of this step is to capture a notion of speaker personality by the types of quote vectors they are likely to produce. Layer 2 maps a sequence of quote vectors $\{\mathbf{q}_1, \ldots, \mathbf{q}_N\}$ to a sequence of inputs $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ for a final softmax regression layer

$$\hat{\mathbf{y}}^{(l)} = \mathrm{softmax}(\mathbf{z}_l \mathbf{W}_s + \mathbf{b}_s). \tag{3}$$

The purpose of Layer 2 is to capture the conversational context surrounding each dialogue line.

Character	Tokenized Line
Fry	dearly beloved , we are here today to remember bender , taken from us in the prime of life , when he was crushed by a runaway semi , driven by the incredible hulk .
Bender	aww , you knew my favourite cause of death !
Fry	now let us each remember the best things about bender in our own way . professor ?
Farnsworth	your standard bending unit is made of an iron- osmium alloy . but bender was different . bender has a point-04 % nickel impurity .
Bender	it 's what made me me .

Table 1: An example dialogue exchange from the *Futurama* episode "A Pharaoh to Remember."

Table 1 shows an example conversation from the *Futurama* corpus. The fourth line from Professor Farnsworth is an example of a quote that could likely be attributed from the corresponding quote vector alone as he is the only character who uses words like "osmium" and "impurity." Bender's lines can only be reliably inferred from conversational context (e.g., the fact that both Fry and the Professor mention Bender by name in their lines). Examples of this sort motivate the two layer approach described in this paper.

3.2.1 Quote Vectors

We consider the following methods for generating quote vectors from arbitrary-length sequences of word vectors.

Simple Averaging (SA)	Basic RNN (B-RNN)
$\mathbf{q}_l = \frac{1}{M_l}\sum_{t=1}^{M_l}\mathbf{w}_t^{(l)}$	$\mathbf{h}_t^{(l)} = \tanh(\mathbf{h}_{t-1}^{(l)}W_1 + \mathbf{w}_t^{(l)}\mathbf{U}_1 + \mathbf{b}_1)$ $\mathbf{q}_l = \mathbf{h}_{M_l}^{(l)}$
GF-RNN [3]	Bidirectional GF-RNN (biGF-RNN)
$\mathbf{h}_t^{(l)} = \mathrm{GRU}(\mathbf{\Sigma}_1, \mathbf{h}_{t-1}^{(l)}, \mathbf{w}_t^{(l)})$ $\mathbf{q}_l = \mathbf{h}_{M_l}^{(l)}$	$\vec{\mathbf{h}}_t^{(l)} = \mathrm{GRU}(\vec{\mathbf{\Sigma}}_1, \vec{\mathbf{h}}_{t-1}^{(l)}, \mathbf{w}_t^{(l)})$ $\overleftarrow{\mathbf{h}}_t^{(l)} = \mathrm{GRU}(\overleftarrow{\mathbf{\Sigma}}_1, \overleftarrow{\mathbf{h}}_{t+1}^{(l)}, \mathbf{w}_t^{(l)})$ $\mathbf{q}_l = [\vec{\mathbf{h}}_{M_l}^{(l)}\ \ \overleftarrow{\mathbf{h}}_0^{(l)}]$

where in all cases the hidden states are initialized to zero, i.e., $\mathbf{h}_0^{(l)} = \vec{\mathbf{h}}_0^{(l)} = \overleftarrow{\mathbf{h}}_{M_l+1}^{(l)} = \mathbf{0}$, and the gated recurrent unit $\mathbf{h}_t = \mathrm{GRU}(\mathbf{\Sigma}, \mathbf{h}_{t-1}, \mathbf{x}_t)$ of Chung et al. [3] is defined with parameters

$\boldsymbol{\Sigma} = (\mathbf{W}^{(z)}, \mathbf{U}^{(z)}, \mathbf{W}^{(r)}, \mathbf{U}^{(r)}, \mathbf{W}, \mathbf{U})$ as:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{x}_t \mathbf{W}^{(z)} + \mathbf{h}_{t-1}\mathbf{U}^{(z)}) \\
\mathbf{r}_t &= \sigma(\mathbf{x}_t \mathbf{W}^{(r)} + \mathbf{h}_{t-1}\mathbf{U}^{(r)}) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{x}_t W + \mathbf{r}_t \circ \mathbf{h}_{t-1}\mathbf{U}) \\
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \tilde{\mathbf{h}}_t.
\end{aligned}$$

3.2.2 Conversational Context

We consider the following methods for generating the $\mathbf{z}_l$ from $\mathbf{q}_l$.

Simple Nonlinearity (SNL)	Basic RNN (B-RNN)
$\mathbf{z}_l = \tanh(\mathbf{q}_l \mathbf{W}_2 + \mathbf{b}_2)$	$\mathbf{z}_l = \tanh(\mathbf{z}_{l-1} W_2 + \mathbf{q}_l \mathrm{U}_2 + \mathbf{b}_1)$
GF-RNN [3]	Bidirectional GF-RNN (biGF-RNN)
$\mathbf{z}_l = \mathrm{GRU}(\boldsymbol{\Sigma}_2, \mathbf{z}_{l-1}, \mathbf{q}_l)$	$\vec{\mathbf{z}}_t = \mathrm{GRU}(\vec{\boldsymbol{\Sigma}}_2, \vec{\mathbf{z}}_{l-1}, \mathbf{q}_l)$ $\overleftarrow{\mathbf{z}}_t = \mathrm{GRU}(\overleftarrow{\boldsymbol{\Sigma}}_2, \overleftarrow{\mathbf{z}}_{l+1}, \mathbf{q}_l)$ $\mathbf{z}_l = [\vec{\mathbf{z}}_l \;\; \overleftarrow{\mathbf{z}}_l]$

We note that the Simple Nonlinearity layer treats each dialogue line independently and does not actually capture any conversational context — it simply provides a layer of non-linearity for use with the Simple Averaging quote vector model. We chose to study the GF-RNN in this work as the design of the gated recurrent unit (GRU) serves as an appropriate proxy for the multiple timescales, i.e., scene and alternating dialogue within a scene, at play in a narrative.

4 Numerical Experiments

4.1 Implementation Details

We implemented the models defined by each (Layer 1, Layer 2) pair using the TensorFlow system [1]. We trained the models to minimize the loss function (1) plus an additional L_2 regularization term $(\lambda/2)\|\mathbf{W}\|_F^2$ for each weight matrix $\mathbf{W}$. We took $\lambda = 10^{-4}$ for all weight matrices at all layers. As an additional form of regularization we applied dropout [10] throughout the models with a keep probability of $p = 0.9$. For recurrent layers of the models, we applied dropout only on the layer inputs and outputs (not the internal state connections) as suggested in [11]. We used the word vector embedding with dimension $d = 100$ (see Equation (2)) from `http://nlp.stanford.edu/data/glove.6B.zip` [9]; the dimensions of the outputs $\mathbf{q}_l$ and $\mathbf{z}_l$ from Layers 1 and 2 respectively are equal to the dimensions of the inputs, unless the layer is biGF-RNN, in which case the output dimension is doubled. We chose the cross-entropy loss weights $\mathbf{w}$ in Equation (1) proportional to the inverse frequency of each speaker's lines in the training corpus. The code for this paper and the tokenized *Futurama* dataset may be found at `https://github.com/schmrlng/RNNQuoteAttribution`.

4.2 Experimental Results

A comparison over model specifications is presented in Table 2, and the confusion matrix for the best-performing model (biGF-RNN, biGF-RNN) is detailed in Table 3. Overall model performance is evaluated according to the weighted cross-entropy loss (1) on the test set of episodes. Concrete predictions for each dialogue line are selected according to the largest value in the prediction vector (Eq. (3)), i.e., $y_{\text{pred}} = \operatorname{argmax} \hat{\mathbf{y}}^{(l)}$. Performance within each class is then evaluated according to the F_1 score, the harmonic mean of precision and recall.

From Table 2 we note that having high quality quote vectors $\mathbf{q}_l$ is the most important factor in determining model success. Indeed, the (B-RNN, SNL) model performs worse than the basic (SA, SNL) model, suggesting that when lines are long (potentially multiple sentences and 100+ tokens) an average over the component word vectors is a better choice than the basic RNN model for semantic summarization. Taking into account conversational context is of secondary concern, but doing so still contributes to a more accurate quote attribution model. This is potentially explained by the idea

Model Specification		Weighted CE Loss		Test F_1 Scores							
Layer 1 (Quote Vectors)	Layer 2 (Conv. Context)	Val Loss	Test Loss	Other (0.46)	Fry (0.66)	Bender (0.69)	Leela (0.74)	Farnsworth (1.04)	Zoidberg (1.34)	Amy (1.50)	Hermes (1.58)
SA	SNL	1.241	1.321	0.573	0.285	0.014	0.206	0.016	0.032	0.055	0.000
SA	B-RNN	1.248	1.335	0.578	0.311	0.027	0.087	0.000	0.000	0.048	0.000
B-RNN	SNL	1.254	1.340	0.534	0.258	0.044	0.136	0.000	0.036	0.096	0.054
SA	GF-RNN	1.223	1.303	0.580	0.259	0.046	0.184	0.047	0.000	0.075	0.000
GF-RNN	SNL	1.206	1.261	0.505	0.344	0.227	0.300	0.172	0.115	0.112	0.043
GF-RNN	GF-RNN	1.182	1.181	0.594	0.377	0.352	0.369	0.310	0.238	0.215	0.000
biGF-RNN	GF-RNN	1.161	1.182	0.580	0.452	0.290	0.350	0.330	0.185	0.158	0.075
GF-RNN	biGF-RNN	1.179	1.185	0.525	0.446	0.330	0.337	0.287	0.217	0.188	0.000
biGF-RNN	biGF-RNN	1.174	1.176	0.534	0.452	0.385	0.323	0.322	0.175	0.183	0.030

Table 2: Model comparison results for the *Futurama* test corpus. The speaker classes included the seven main crew members and an eighth class consisting of all other speakers. Cross-entropy loss weights (see Equation (1)) are given below the names of each character. Compared to Bender, it seems that Leela has a particularly distinctive speaking style, as even though she speaks less, most of the models are able to detect her lines reasonably accurately.

that speakers are identified most clearly through the personality of their speech; we fall back on conversational context and positive identifications of surrounding characters if our initial impression of the quote leaves us unsure. Introducing bidirectionality increases the prediction accuracy slightly (again this holds particularly true for Layer 1) and the best model by test loss is (biGF-RNN, biGF-RNN).

	Predicted Speaker									
Actual Speaker	Other	Fry	Bender	Leela	Farnsworth	Zoidberg	Amy	Hermes	Precision	Recall
Other	350	69	44	91	74	9	25	16	0.552	0.516
Fry	58	138	23	43	13	6	13	4	0.440	0.463
Bender	88	43	87	35	12	4	6	4	0.502	0.311
Leela	60	29	8	82	13	2	21	2	0.282	0.377
Farnsworth	28	11	4	12	46	4	3	2	0.261	0.418
Zoidberg	17	8	1	4	7	7	1	2	0.212	0.148
Amy	23	12	1	13	5	1	14	0	0.166	0.202
Hermes	10	3	5	10	6	0	1	1	0.032	0.027

Table 3: Detailed test set results for the (biGF-RNN, biGF-RNN) model. The confusion matrix showing counts of actual speaker (by row) vs. the model's predicted speaker (by column) is on the left side, and the precision and recall statistics for each class are on the right side.

From Table 3 it is clear that the "Other" class dominates the predictions. This is potentially due to the existence of many one-off, single-episode characters in *Futurama* with varied personalities that can mimic any of the main crew members. The plurality of "Other" characters also affects the ability of the model to learn conversational flow, as some conversations are dominated by multiple "Other" characters. In Table 4 we show test results for predicting the speaker of lines known a priori to be spoken by the seven main crew members. The model is near-50% accurate in identifying each of the four most common speakers, but has a very difficult time identifying Hermes, who throughout the corpus only speaks $1/6$ as much as Fry.

5 Conclusions

We have shown that using gated-feedback recurrent neural networks for both quote summarization and conversation following provides improved performance over simpler neural network models at

	Predicted Speaker									
Actual Speaker	Fry	Bender	Leela	Farnsworth	Zoidberg	Amy	Hermes	Precision	Recall	F_1 Score
Fry	162	31	55	17	8	16	9	0.543	0.543	0.543
Bender	49	127	50	20	5	12	16	0.648	0.455	0.534
Leela	39	17	103	23	3	29	3	0.384	0.474	0.424
Farnsworth	14	8	18	53	5	5	7	0.392	0.481	0.432
Zoidberg	12	3	10	7	7	2	6	0.233	0.148	0.181
Amy	18	3	19	7	2	19	1	0.223	0.275	0.246
Hermes	4	7	13	8	0	2	2	0.045	0.055	0.050

Table 4: Test set results for the (biGF-RNN, biGF-RNN) model when predictions were taken as the argmax over the prediction vector $\hat{\mathbf{y}}$ restricted to non-"Other" characters; the counts for lines truly spoken by "Other" characters are omitted.

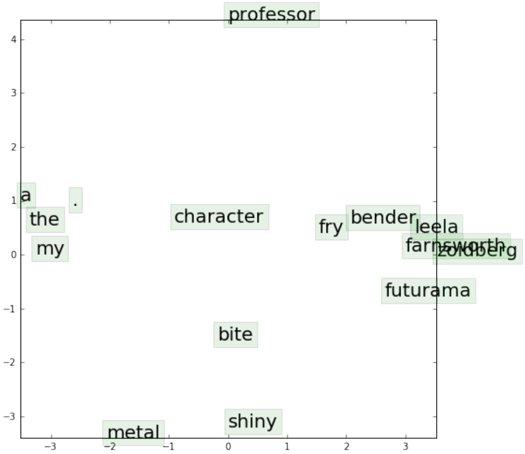

Figure 1: *Futurama* character word vectors visualized with respect to a selection of other words.

the task of dialogue-only quote attribution for screenplays and novels. Yet there is clearly still much room for improvement. Alternative methods of quote summarization, e.g., [6], may provide more accurate windows into speaker personality. Allowing for selective retraining of word vectors, in particular for tokens corresponding to speaker names, may also lead to better performance. Figure 1 shows the GloVe word vectors for a selection of character tokens (plus a few additional words) plotted by their components along the vectors' first two principal axes. In the context of the Gigaword5 + Wikipedia2014 corpus, all of the *Futurama* characters are essentially equivalent. For the purpose of quote attribution, however, we'd like the word "bender" in the exchange from Table 1 to send a strong signal that the next speaker is likely to be Bender. We expect that such word vector retraining would increase the benefits provided by using a recursive neural network in Layer 2 (conversational context).

Besides improving the performance of supervised dialogue-only quote attribution, however, the model specifications described in this paper might be useful in future work for unsupervised clustering of speakers. Indeed, this two layer outline can be used to model sequential interaction between any underlying generative models of data sequences, not just characters in narratives producing words of dialogue. In particular the author of this paper was motivated in this work by the idea of using recursive neural networks to model robotic interactions, where the actions ("speech") of an agent are highly determined by its intent ("character") as well as its ongoing interaction ("conversation") with other agents in its environment.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Good-

fellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Arun Chaganty and Grace Muzny. Quote attribution for literary text with neural networks. Avaliable at `http://cs224d.stanford.edu/reports/ChagantyArun.pdf`.

[3] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. *arXiv preprint arXiv:1502.02367*, 2015.

[4] David K. Elson and Kathleen R. McKeown. Automatic attribution of quoted speech in literary narrative. In *AAAI*. Citeseer, 2010.

[5] Hua He, Denilson Barbosa, and Grzegorz Kondrak. Identification of speakers in novels. In *ACL (1)*, pages 1312–1320, 2013.

[6] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[7] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[8] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3, 2010.

[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[11] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.