# Incremental Text Classification

*Alon Albalak*

University of California, Santa Barbara

**Abstract**

This paper discusses text classification with increasing information as a form of context understanding. We consider 5 different methods of sequence representation. 1 statistical method: Tf-Idf; 2 simple neural models: Doc2Vec, Universal Sentence Encoder; and 2 hierarchical models: Doc2Vec+FastText, Universal Sentence Encoder+Doc2Vec. We analyze these classification methods on the DBPedia ontology dataset with the aim of setting a baseline measure for future study of dynamic context understanding.

## 1 Introduction

**Context Understanding** Classification is one of the fundamental tasks in machine learning. Specifically in natural language processing, classification arises in many forms such as named entity recognition, word sense disambiguation, and sentiment analysis. Classifying segments of text into categories is at the basis of natural language understanding. Before acheiving natural language understanding machines should first be able to identify subjects, such as names or places, in a text and in order to have a conversation they must be able to identify the topic of discussion. More concisely, machines must understand context.

**Text Classification** The first techniques developed for text classification often involved designing a set of hand-written rules and evaluating the text using statistical models. However, along with the significant increase in data available since the establishment of the internet, discriminative models and specifically neural network models have seen a similar increase in use.

Traditionally, text information has come in the form of single-author data such as books or articles. However, with the increasing use of chatbots there has been a surge of multi-author text and in particular, dialogue. One aspect of dialogue that is seen much less frequently in single-author than multi-author text is a dynamic topic.

The incrementality of spoken word is much more obvious than in written text. In written NLP tasks, processing usually takes an entire document as input, and is often a text written by a single author. For speech tasks though, the data is often an interaction between multiple people, and so there has been much more research done involving incremental speech data [1] [2] [3] [4]. This has left a lack of incremental classification data when it comes to written text.

## 2 Related Work

**Incremental Speech Processing** Research into incremental speech processing has existed for nearly 50 years, and has changed drastically, starting with the statistical methods used in the 1970s [5] to the many recurrent neural network based models of today [6]. Due to the inherent incrementality of speech understanding and processing, there has been much work done in a variety of incremental speech based tasks such as speech recognition[7] [1], natural language generation [8], and dialogue state tracking [3] [4].

**Incremental Text Processing** In addition to research in incremental speech processing, there has also been research into incremental text processing. In particular, machine translation [9], question answering [10], and information status classification [11].

Until the early 2010s most text classification techniques were statistical in nature, such as the methods based on ICA, LSA [12], and Tf-Idf [13]. However, since then there has been a shift towards neural network based models. In 2013 Mikolov et al. [14] designed their distributed vector representation of words (word2vec), and only a year later they had already published a similar model for variable length sequences, the Distributed Memory Model of Paragraph Vectors (Doc2Vec) [15]. One of the benefits of creating fixed length embeddings is that they allow for further downstream uses in natural language processing tasks. The development of more advanced neural network architectures has led to other methods of text classification using LSTM [16], CNN [17], and character-level CNN [18]. Even more recently, starting in 2017, techniques using supervised transfer learning to learn

universal embeddings became popular with promising results. Models such as ULMFiT [19] and Universal Sentence Embeddings [20] have shown good results in text classification tasks.

**Incremental Text Classification** Although there have been significant amounts of research in these areas, there does not seem to be any previous work into incremental text classification. This paper aims to create a baseline for incremental text classification to allow future studies into dynamic context understanding.

# 3 My Approach

Here we will provide a brief discussion of the methods used.

**Tf-Idf** Term Frequency-Inverse Document Frequency is a method used to weigh the importance of a term in a particular document [21]. In this experiment, we use the bag of words model for the vocabulary. The term frequency of a word is simply its raw count in a particular document. The inverse document frequency of a word is the inverse of the number of documents it appears in. This represents a type of word specificity, penalizing words which occur in many documents and increasing the weight of a less frequent word.

**Doc2Vec** Doc2Vec is a neural model, trained in much the same methods as word2vec [15] [14]. Where word2vec can be trained in either the continuous bag of words, or the skip-gram architectures, Doc2Vec can also be trained on either a single word or multiple word prediction task. The single word prediction task takes a few words plus a representation of the paragraph in one-hot form. This form is called the Distributed Memory Model of Paragraph Vectors. The Distributed Bag of Words version of Paragraph Vector takes only the paragraph i.d. as input, and predicts randomly sampled words from the paragraph. Also similarly to word2vec, the weights used to create the embedding layer are saved after training to be used to create dense vector representations, but for variable length sequences. For this experiment, we use the distributed bag of words version.

**FastText** FastText is another neural model also similar to word2vec. Instead of a continuous bag of words architecture though, it uses a bag of n-grams approach in order to maintain some of the local word order [22].

**Universal Sentence Encoder** The universal sentence encoder(USE) was specifically developed to target transfer learning to other natural language processing tasks with the goal of being as general purpose as possible [20]. In order to create a general purpose sentence encoding, the output embedding from a single encoding model is fed into multiple downstream tasks. Cer et al. designed 2 models which can create sentence embeddings, one based on the transformer model, and another using a deep averaging network. For this experiment, we use the transformer encoder version because between the 2 models it was shown to have the higher task transfer performance.

# 4 Dataset

The DBpedia ontology dataset has 560,000 training samples and 70,000 testing samples of 14 non-overlapping categories from DBpedia. In models where hyperparameters need to be tuned, 5,000 training examples were pulled from each category to be used as a validation set totaling 70,000 samples in size. Each sample in the dataset is an abstract of an article from Wikipedia along with its class.

# 5 Experimental Settings

In order to test incrementality, we simply train and test on increasing amounts of data. This means that we consider the first word of an abstract to be a single sample, and consider the first 2 words to be another sample, etc. until we have included the entire abstract.
In this experiment, we use 5 different methods of classification. We have 1 statistical method, and the remaining methods all involve neural models, however we split the neural models into 2 categories: simple and hierarchical models.
For the simple models, we test using only Doc2Vec and Universal Sentence Encoder. For the hierarchical models, we have one technique creating embeddings at the document level, and the other technique outputting embeddings for either the sentence or word level.

**Tf-Idf** The only statistical method that was tested in this experiment is a bag-of-words and its Tf-Idf [21] using word count as features. Before getting word counts, all words were converted to lowercase and a porter stemmer was used [23]. Tf-Idf was used as the input feature for a multinomial naive bayes classifier [24].

**Doc2Vec** The Doc2Vec embeddings were trained on only the training data, where any word which appeared only once in the data was disregarded. After training the Doc2Vec module, a very brief search for architecture was implemented by evaluating text classification on whole abstracts using the training and validation datasets. The best architecture took the 300 dimensional output of Doc2Vec into two 128-dimension fully connected layers using the ReLU non-linearity and then into a softmax classifier. The loss was minimized with the Nesterov Adam optimizer using the standard parameters.

**Universal Sentence Encoder** The 512-dimensional output from the pretrained USE model was fed into a 256-dimension fully connected layer with ReLU non-linearity, which was fed into a softmax classifier. Again, we minimize loss with the Nesterov Adam optimizer.

**Doc2Vec + FastText** In this hierarchical model, a pretrained FastText model was used. The FastText module creates a 300 dimensional word embedding for only the last word in the current sequence. The FastText embedding is fed into a fully connected layer with 32 neurons and uses a ReLU non-linearity. In addition to the FastText module, we use the same Doc2Vec model as prior. However, we take the output from the last fully connected layer of the Doc2Vec model and concatenate it with the FastText layer, and together those are classified by a softmax layer.

**Universal Sentence Encoder + Doc2Vec** In this hierarchical model we use the same architectures from the individual USE and Doc2Vec models, except that we concatenate the outputs from the fully connected layers and those are together classified with a softmax layer. In this experiment, the input to the USE model is the entire article and the input to the Doc2Vec model is only the most recent sentence.

# 6 Metrics

For this experiment, only accuracy was considered. Since we are testing for incremental classification, we look at the classification accuracy over sentence length.

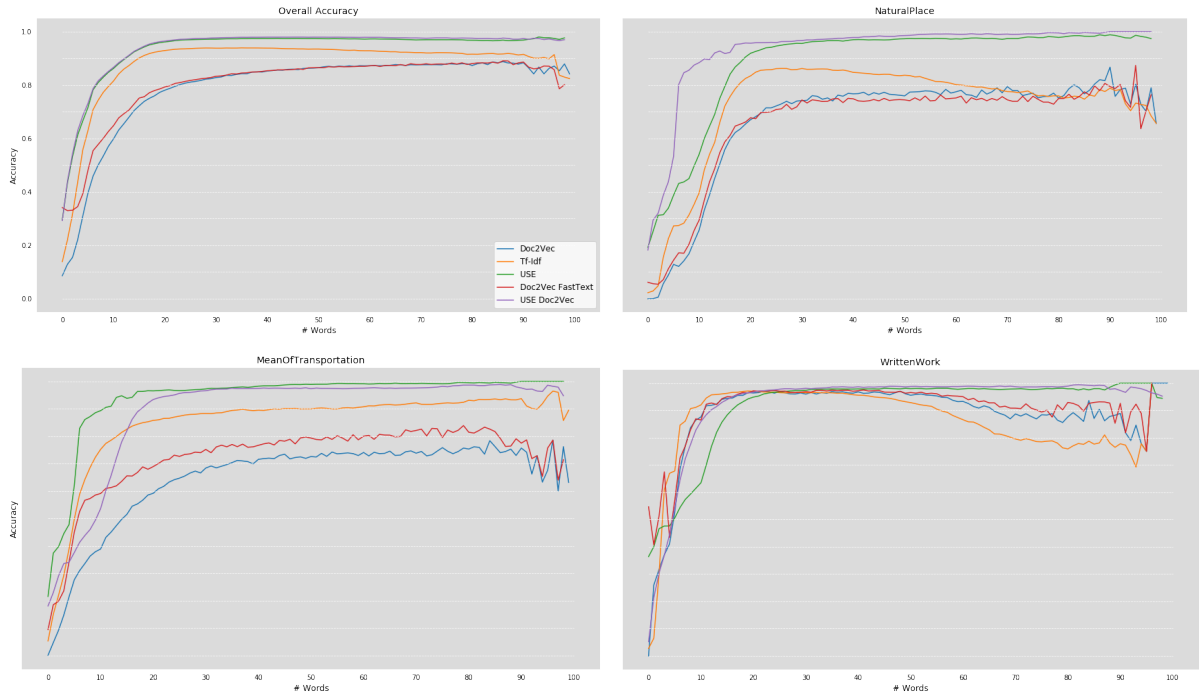


**Figure 1:** Accuracy vs. Sentence Length for Particular Classes. Clock-wise from top left: Overall, "Natural Places","Written Work","Means of Transportation"

| Method | Word | Accuracy |
|---|---|---|
| Doc2Vec | 87 | 0.887392 |
| Tfidf | 36 | 0.938836 |
| USE | 94 | 0.979702 |
| Doc2Vec_FastText | 87 | 0.890567 |
| USE_Doc2Vec | 47 | 0.979256 |

**Figure 2:** Peak accuracy achieved by each model, and # of words at which the accuracy occured. Largest values highlighted in green, smallest values in red

# 7 Experimental Results

Overall, we see in figure 1 that the USE and USE+Doc2Vec models perform best, and they have very similar accuracies. One thing of note is that although USE, USE+Doc2Vec, and Tf-Idf all get above 90% accuracy, Tf-Idf actually peaks at 36 words and slowly loses accuracy for longer sequences while both USE and USE+Doc2Vec have stable accuracies for sequences greater than 20 words. Also, USE alone actually increases in accuracy with increased sequence length, as shown in figure 2, while the combination of USE+Doc2Vec peaks at 47.

In addition, neither hierarchical model has a significant gain in peak accuracy over its simple counterpart. Both hierarchical models have less than 1% improvement over their corresponding simple model. At smaller sequence lengths though, Doc2Vec+FastText model does see increased accuracy over the simple Doc2Vec model.

Interestingly, in some individual classes, such as Natural Place, we do see that Doc2Vec gives USE a significant boost in early accuracy. In contrast, in Means of Transportation, the Doc2Vec model does relatively poorly, and appears to negatively influence the USE+Doc2Vec model. Finally, in Written Work, all models have a very similar peak accuracy, but it is very clear that Doc2Vec and Tf-Idf are unable to properly classify the longer sequences in this class.

# 8 Future Work

**Other Methods** There are a large number of other methods that could have been tested, but we felt these to be a good variety for preliminary testing considering time and computing constraints. For future research, it would be interesting to see how other sentence encodings compare, especially LSTM based encodings.

**Dynamic Context** The task of carrying conversation is inherently dynamic since through the course of a conversation the subject of discussion may change many times. Using these results as a basis, it would be interesting to see how well these and other techniques can keep up with changing topics.

# 9 Conclusion

We present in this paper the novel task of incremental text classification, and show accuracy of 5 different methods in experimental results.

# References


[1] K. Sagae, G. Christian, D. DeVault, and D. R. Traum, "Towards natural language understanding of partial speech recognition results in dialogue systems," in *HLT-NAACL*, 2009.

[2] S. C. Stoness, J. Allen, G. Aist, and M. Swift, "Using real-world reference to improve spoken language understanding," in *AAAI*, 2005.

[3] W. Chen, J. Chen, Y. Su, X. Wang, D. Yu, X. Yan, and W. Y. Wang, "XL-NBT: A Cross-lingual Neural Belief Tracking Framework," *arXiv e-prints*, p. arXiv:1808.06244, Aug. 2018.

[4] V. Petukhova and H. Bunt, "Incremental dialogue act understanding."

[5] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[6] A. Köhn, "Incremental natural language processing: Challenges, strategies, and evaluation," *CoRR*, vol. abs/1805.12518, 2018.

[7] E. Selfridge, I. Arizmendi, P. A. Heeman, and J. D. Williams, "Stability and accuracy in incremental speech recognition," in *SIGDIAL Conference*, 2011.

[8] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in conversational systems," *Computer Speech & Language*, vol. 27, no. 1, pp. 243 – 262, 2013. Special issue on Paralinguistics in Naturalistic Speech and Language.

[9] J. Gu, G. Neubig, K. Cho, and V. O. K. Li, "Learning to translate in real-time with neural machine translation," *CoRR*, vol. abs/1610.00388, 2016.

[10] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, "A neural network for factoid question answering over paragraphs," in *Empirical Methods in Natural Language Processing*, 2014.

[11] Y. Hou, "Incremental fine-grained information status classification using attention-based lstms," in *COLING*, 2016.

[12] Q. Pu and G.-W. Yang, "Short-text classification based on ica and lsa," in *Advances in Neural Networks - ISNN 2006* (J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, and H. Yin, eds.), (Berlin, Heidelberg), pp. 265–270, Springer Berlin Heidelberg, 2006.

[13] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of tf*idf, lsi and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758 – 2765, 2011.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[15] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014.

[16] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using lstm for region embeddings," *arXiv preprint arXiv:1602.02373*, 2016.

[17] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun, "Very deep convolutional networks for natural language processing," *CoRR*, vol. abs/1606.01781, 2016.

[18] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *CoRR*, vol. abs/1509.01626, 2015.

[19] J. Howard and S. Ruder, "Fine-tuned language models for text classification," *CoRR*, vol. abs/1801.06146, 2018.

[20] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018.

[21] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, Nov. 1975.

[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016.

[23] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[24] A. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, *Multinomial Naive Bayes for Text Classification Revisited*, ch. 43, pp. 488–499. Springer, 2004.