#TAGSPACE: Semantic Embeddings from Hashtags

Jason Weston
Facebook AI Research
jase@fb.com

Sumit Chopra
Facebook AI Research
spchopra@fb.com

Keith Adams
Facebook AI Research
kma@fb.com

Abstract

We describe a convolutional neural network that learns feature representations for short textual posts using hashtags as a supervised signal. The proposed approach is trained on up to 5.5 billion words predicting 100,000 possible hashtags. As well as strong performance on the hashtag prediction task itself, we show that its learned representation of text (ignoring the hashtag labels) is useful for other tasks as well. To that end, we present results on a document recommendation task, where it also outperforms a number of baselines.

1 Introduction

Hashtags (single tokens often composed of natural language n-grams or abbreviations, prefixed with the character '#') are ubiquitous on social networking services, particularly in short textual documents (a.k.a. posts). Authors use hashtags to diverse ends, many of which can be seen as labels for classical NLP tasks: disambiguation (`chips #futurism` vs. `chips #junkfood`); identification of named entities (`#sf49ers`); sentiment (`#dislike`); and topic annotation (`#yoga`). *Hashtag prediction* is the task of mapping text to its accompanying hashtags. In this work we propose a novel model for hashtag prediction, and show that this task is also a useful surrogate for learning good representations of text.

Latent representations, or *embeddings*, are vectorial representations of words or documents, traditionally learned in an unsupervised manner over large corpora. For example LSA (Deerwester et al., 1990) and its variants, and more recent neural-network inspired methods like those of Bengio et al. (2006), Collobert et al. (2011) and word2vec (Mikolov et al., 2013) learn word embeddings. In the word embedding paradigm, each word is represented as a vector in $\mathbb{R}^n$, where n is a hyper-parameter that controls capacity. The embeddings of words comprising a text are combined using a model-dependent, possibly learned function, producing a point in the same embedding space. A similarity measure (for example, inner product) gauges the pairwise relevance of points in the embedding space.

Unsupervised word embedding methods train with a reconstruction objective in which the embeddings are used to predict the original text. For example, word2vec tries to predict all the words in the document, given the embeddings of surrounding words. We argue that hashtag prediction provides a more direct form of supervision: the tags are a labeling by the author of the salient aspects of the text. Hence, predicting them may provide stronger semantic guidance than unsupervised learning alone. The abundance of hashtags in real posts provides a huge labeled dataset for learning potentially sophisticated models.

In this work we develop a convolutional network for large scale ranking tasks, and apply it to hashtag prediction. Our model represents both words and the entire textual post as embeddings as intermediate steps. We show that our method outperforms existing unsupervised (word2vec) and supervised (WSABIE (Weston et al., 2011)) embedding methods, and other baselines, at the hashtag prediction task.

We then probe our model's generality, by transfering its learned representations to the task of *personalized document recommendation*: for each of M users, given N previous positive interactions with documents (likes, clicks, etc.), predict the $N+1$'th document the user will positively interact with. To perform well on this task, the representation should capture the user's interest in textual content. We find representations trained on hashtag prediction outperform representations from unsupervised learning, and that our convolu-

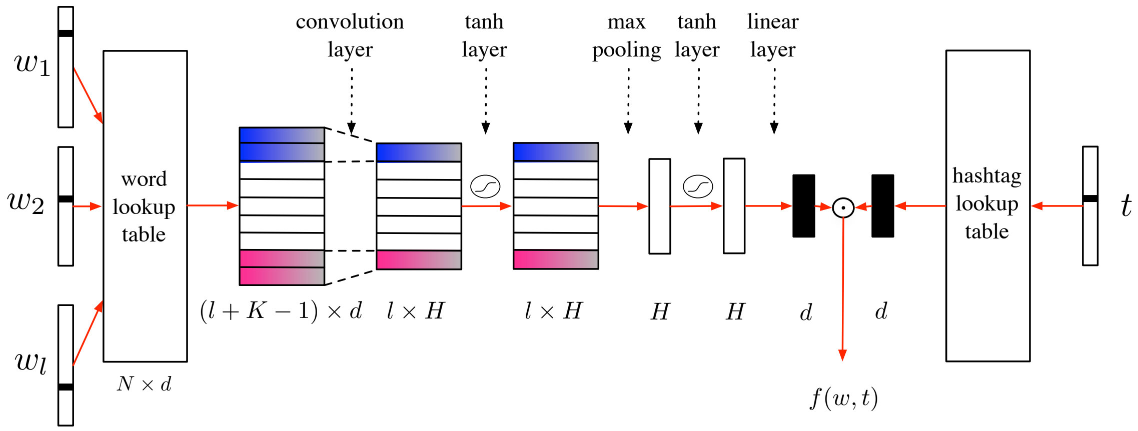

Figure 1: #TagSpace convolutional network $f(w,t)$ for scoring a (document, hashtag) pair.

tional architecture performs better than Wsabie trained on the same hashtag task.

2 Prior Work

Some previous work (Davidov et al., 2010; Godin et al., 2013; She and Chen, 2014) has addressed hashtag prediction. Most such work applies to much smaller sets of hashtags than the 100,000 we consider, with the notable exception of Ding et al. (2012), which uses an unsupervised method.

As mentioned in Section 1, many approaches learn unsupervised word embeddings. In our experiments we use word2vec (Mikolov et al., 2013) as a representative scalable model for unsupervised embeddings. Wsabie (Weston et al., 2011) is a supervised embedding approach that has shown promise in NLP tasks (Weston et al., 2013; Hermann et al., 2014). Wsabie is shallow, linear, and ignores word order information, and so may have less modeling power than our approach.

Convolutional neural networks (CNNs), in which shared weights are applied across the input, are popular in the vision domain and have recently been applied to semantic role labeling (Collobert et al., 2011) and parsing (Collobert, 2011). Neural networks in general have also been applied to part-of-speech tagging, chunking, named entity recognition (Collobert et al., 2011; Turian et al., 2010), and sentiment detection (Socher et al., 2013). All these tasks involve predicting a limited (2-30) number of labels. In this work, we make use of CNNs, but apply them to the task of ranking a very large set of tags. We thus propose a model and training scheme that can scale to this class of problem.

3 Convolutional Embedding Model

Our model #TagSpace (see Figure 1), like other word embedding models, starts by assigning a d-dimensional vector to each of the l words of an input document $w_1, \ldots, w_l$, resulting in a matrix of size $l \times d$. This is achieved using a matrix of $N \times d$ parameters, termed the lookup-table layer (Collobert et al., 2011), where N is the vocabulary size. In this work N is 10^6, and each row of the matrix represents one of the million most frequent words in the training corpus.

A convolution layer is then applied to the $l \times d$ input matrix, which considers all successive windows of text of size K, sliding over the document from position 1 to l. This requires a further $Kd \times H$ weights and H biases to be learned. To account for words at the two boundaries of the document we also apply a special padding vector at both ends. In our experiments K was set to 5 and H was set to 1000. After the convolutional step, a $\tanh$ nonlinearity followed by a $\max$ operation over the $l \times H$ features extracts a fixed-size (H-dimensional) global feature vector, which is independent of document size. Finally, another $\tanh$ non-linearity followed by a fully connected linear layer of size $H \times d$ is applied to represent the entire document in the original embedding space of d-dimensions.

Hashtags are also represented using d-dimensional embeddings using a lookup-table. We represent the top 100,000 most frequent tags. For a given document w we then rank any given hashtag t using the scoring function:

$$f(w,t) = e_{conv}(w) \cdot e_{lt}(t)$$

where $e_{conv}(w)$ is the embedding of the document by the CNN just described and $e_{lt}(t)$ is the embedding of a candidate tag t. We can thus rank all candidate hashtags via their scores $f(w,t)$, largest first.

To train the above scoring function, and hence the parameters of the model we minimize a ranking loss similar to the one used in WSABIE as a training objective: for each training example, we sample a positive tag, compute $f(w,t^+)$, then sample random tags $\bar{t}$ up to 1000 times until $f(w,\bar{t}) > m + f(w,t^+)$, where m is the margin. A gradient step is then made to optimize the pairwise hinge loss:

$$\mathcal{L} = \max\{0, m - f(w,t^+) + f(w,\bar{t})\}.$$

We use $m = 0.1$ in our experiments. This loss function is referred to as the WARP loss in (Weston et al., 2011) and is used to approximately optimizing the top of the ranked list, useful for metrics like precision and recall@k. In particular, the search for a negative candidate tag means that more energy is spent on improving the ranking performance of positive labels already near the top of the ranked list, compared to only randomly sampling of negatives, which would optimize the average rank instead.

Minimizing our loss is achieved with parallel stochastic gradient descent using the hogwild algorithm (Niu et al., 2011). The lookup-table layers are initialized with the embeddings learned by WSABIE to expedite convergence. This kind of 'pre-training' is a standard trick in the neural network literature, see e.g. (Socher et al., 2011).

The ranking loss makes our model scalable to 100,000 (or more) hashtags. At each training example only a subset of tags have to be computed, so it is far more efficient than a standard classification loss that considers them all.

4 Experiments

4.1 Data

Our experiments use two large corpora of posts containing hashtags from a popular social network.[1] The first corpus, which we call *people*, consists of 201 million posts from individual user accounts, comprising 5.5 billion words.

The second corpus, which we call *pages*, consists of 35.3 million page posts, comprising 1.6 billion words. These posts' authorial voice is a public entity, such as a business, celebrity, brand, or product. The posts in the pages dataset are presumably intended for a wider, more general audience than the posts in the people dataset. Both are summarized in Table 1.

Dataset	Posts (millions)	Words (billions)	Top 4 tags
Pages	35.3	1.6	#fitness, #beauty, #luxury, #cars
People	201	5.5	#FacebookIs10, #love, #tbt, #happy

Table 1: Datasets used in hashtag prediction.

Both corpora comprise posts between February 1st and February 17th, 2014. Since we are not attempting a multi-language model, we use a simple trigram-based language prediction model to consider only posts whose most likely language is English.

The two datasets use hashtags very differently. The pages dataset has a fatter head, with popular tags covering more examples. The people dataset uses obscure tags more heavily. For example, the top 100 tags account for 33.9% of page tags, but only 13.1% of people tags.

4.2 Hashtag prediction

The hashtag prediction task attempts to rank a post's ground-truth hashtags higher than hashtags it does not contain. We trained models on both the people and page datasets, and collected precision at 1, recall at 10, and mean rank for 50,000 randomly selected posts withheld from training. A further 50,000 withheld posts are used for selecting hyperparameters. We compare #TAGSPACE with the following models:

Frequency This simple baseline ignores input text, always ranking hashtags by their frequency in the training data.

#words This baseline assigns each tag a static score based on its frequency plus a large bonus if it corresponds to a word in the input text. For example, on input "crazy commute this am", #words ranks `#crazy`, `#commute`, `#this` and `#am` highest, in frequency order.

Word2vec We trained the unsupervised model of Mikolov et al. (2013) on both datasets, treating hashtags the same as all other words. To ap-

[1] Both corpora were de-identified during collection.

Crazy commute this am, was lucky to even get in to work.	#nyc, #snow, #puremichigan, #snowday, #snowstorm, #tubestrike, #blizzard, #commute, #snowpocalypse, #chiberia
This can't go on anymore, we need marriage equality now!	#samelove, #equalrights, #equality, #equalityforall, #loveislove, #lgbt, #marriageequality, #noh8, #gayrights, #gaymarriage
Kevin spacey what a super hottie :)	#houseofcards, #hoc, #houseofcardsseason2, #season2, #kevinspacey, #frankunderwood, #netflix, #suits, #swoon, #hubbahubba
Went shopping today and found a really good place to get fresh mango.	#mango, #shopping, #heaven, #100happydays, #yummy, #lunch, #retailtherapy, #yum, #cravings, #wholefoods
Went running today -- my feet hurt so much!	#running, #ouch, #pain, #nopainnogain, #nike #marathontraining, #sore, #outofshape, #nikeplus, #runnerproblems
Wow, what a goal that was, just too fast, Mesut Ozil is the best!	#arsenal, #coyg, #ozil, #afc, #arsenalfc #lfc, #ynwa, #mesut, #gunners, #ucl
Working really hard on the paper all last night.	#thestruggle, #smh, #lol, #collegelife, #homework #sad, #wtf, #confused, #stressed, #work
The restaurant was too expensive and the service was slow.	#ripoff, #firstworldproblems, #smh, #fail, #justsaying #restaurant, #badservice, #food, #middleclassproblems, #neveragain
The restaurant had great food and was reasonably priced.	#dinner, #restaurant, #yum, #food, #delicious #stuffed, #goodtimes, #foodporn, #yummy, #winning
He has the longest whiskers, omg so sweet!	#cat, #kitty, #meow, #cats, #catsofinstagram #crazycatlady, #cute, #kitten, #catlady, #adorable

Table 2: #TAGSPACE (256 dim) predictions for some example posts.

ply these word embeddings to ranking, we first sum the embeddings of each word in the text (as word2vec does), and then rank hashtags by similarity of their embedding to that of the text.[2]

WSABIE (Weston et al., 2011) is a supervised bilinear embedding model. Each word and tag has an embedding. The words in a text are averaged to produce an embedding of the text, and hashtags are ranked by similarity to the text embedding. That is, the model is of the form:

$$f(w,t) = w^\top U^\top V t$$

where the post w is represented as a bag of words (a sparse vector in $\mathbb{R}^N$), the tag is a one-hot-vector in $\mathbb{R}^N$, and U and V are $k \times N$ embedding matrices. The WARP loss, as described in section 3, is used for training.

Performance of all these models at hashtag prediction is summarized in Tables 3 and 4. We find similar results for both datasets. The frequency and #words baselines perform poorly across the board, establishing the need to learn from text. Among the learning models, the unsupervised word2vec performs the worst. We believe this is due to it being unsupervised – adding supervision better optimizes the metric we evaluate. #TAGSPACE outperforms WSABIE at all dimensionalities. Due to the relatively large test sets, the results are statistically significant; for example, comparing #TAGSPACE (64 dim) beats Wsabie (64 dim) for the page dataset 56% of the time, and draws 23% of the time in terms of the rank metric, and is statistically significant with a Wilcoxon signed-rank test.

Some example predictions for #TAGSPACE are given for some constructed examples in Table 2. We also show nearest word embeddings to the posts. Training data was collected at the time of the pax winter storm, explaining predictions for the first post, and Kevin Spacey appears in the show "House of Cards,". In all cases the hashtags reveal labels that capture the semantics of the posts, not just syntactic similarity of individual words.

Comparison to Production System We also compare to a proprietary system in production in Facebook for hashtag prediction. It trains a logistic regression model for every hashtag, using a bag of unigrams, bigrams, and trigrams as the

[2] Note that the unsupervised Word2vec embeddings could be used as input to a supervised classifier, which we did not do. For a supervised embedding baseline we instead use WSABIE. WSABIE trains word embeddings U and hashtag embeddings V in a supervised fashion, whereas Word2vec trains them both unsupervised. Adding supervision to Word2vec would effectively do something in-between: U would still be unsupervised, but V would then be supervised.

Method	dim	P@1	R@10	Rank
Freq. baseline	-	1.06%	2.48%	11277
#words baseline	-	0.90%	3.01%	11034
Word2Vec	256	1.21%	2.85%	9973
Word2Vec	512	1.14%	2.93%	8727
Wsabie	64	4.55%	8.80%	6921
Wsabie	128	5.25%	9.33%	6208
Wsabie	256	5.66%	10.34%	5519
Wsabie	512	5.92%	10.74%	5452
#TagSpace	64	6.69%	12.42%	3569
#TagSpace	128	6.91%	12.57%	3858
#TagSpace	256	**7.37%**	**12.58%**	**3820**

Table 3: Hashtag test results for people dataset.

Method	dim	P@1	R@10	Rank
Freq. baseline	-	4.20%	1.59%	11103
#words baseline	-	2.63%	5.05%	10581
Word2Vec	256	4.66%	8.15%	10149
Word2Vec	512	5.26%	9.33%	9800
Wsabie	64	24.45%	29.64%	2619
Wsabie	128	27.47%	32.94%	2325
Wsabie	256	29.76%	35.28%	1992
Wsabie	512	30.90%	36.96%	1184
#TagSpace	64	34.08%	38.96%	1184
#TagSpace	128	36.27%	41.42%	1165
#TagSpace	256	**37.42%**	**43.01%**	**1155**

Table 4: Hashtag test results for pages dataset.

input features. Unlike the other models we consider here, this baseline has been trained using a set of approximately 10 million posts. Engineering constraints prevent measuring mean rank performance. We present it here as a serious effort at solving the same problem from outside the embedding paradigm. On the people dataset this system achieves 3.47% P@1 and 5.33% R@10. On the pages dataset it obtains 5.97% P@1 and 6.30% R@10. It is thus outperformed by our method. However, we note the differences in experimental setting mean this comparison is perhaps not completely fair (different training sets). We expect performance of linear models such as this to be similar to Wsabie as that has been in the case in other datasets (Gupta et al., 2014), but at the cost of more memory usage. Note that models like logistic regression and SVM do not scale well if you have millions of hashtags, which we could handle in our models.

4.3 Personalized document recommendation

To investigate the generality of these learned representations, we apply them to the task of recommending documents to users based on the user's interaction history. The data for this task comprise anonymized day-long interaction histories for a tiny subset of people on a popular social networking service. For each of the 34 thousand people considered, we collected the text of between 5 and 167 posts that she has expressed previous positive interactions with (likes, clicks, etc.). Given the person's trailing $n-1$ posts, we use our models to predict the n'th post by ranking it against 10,000 other unrelated posts, and measuring precison and recall. The score of the n^{th} post is obtained by taking the max similarity over the $n-1$ posts. We use cosine similarity between post embeddings instead of the inner product that was used for hashtag training so that the scores are not unduly influenced by document length. All learned hashtag models were trained on the people dataset. We also consider a TF-IDF weighted bag-of-words baseline (BoW). The results are given in Table 5.

Method	dim	P@1	R@10	R@50
Word2Vec	256	0.75%	1.96%	3.82%
BoW	-	1.36%	4.29%	8.03%
Wsabie	64	0.98%	3.14%	6.65%
Wsabie	128	1.02%	3.30%	6.71%
Wsabie	256	1.01%	2.98%	5.99%
Wsabie	512	1.01%	2.76%	5.19%
#TagSpace	64	1.27%	4.56%	9.64%
#TagSpace	128	1.48%	4.74%	9.96%
#TagSpace	256	**1.66%**	**5.29%**	**10.69%**
Wsabie+ BoW	64	1.61%	4.83%	9.00%
#TagSpace+ BoW	64	1.80%	5.90%	11.22%
#TagSpace+ BoW	256	**1.92%**	**6.15%**	**11.53%**

Table 5: Document recommendation task results.

Hashtag-based embeddings outperform BoW and unsupervised embeddings across the board, and #TagSpace outperforms Wsabie. The best results come from summing the bag-of-words scores with those of #TagSpace.

5 Conclusion

#TagSpace is a convolutional neural network that learns to rank hashtags with a minimum of task-specific assumptions and engineering. It performs well, beating several baselines and an industrial system engineered for hashtag prediction. The semantics of hashtags cause #TagSpace to learn a representation that captures many salient aspects of text. This representation is general enough to port to the task of personalized document recommendation, where it outperforms other well-known representations.

Acknowledgements

We thank Ledell Wu and Jeff Pasternak for their help with datasets and baselines.

References

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In *International Conference on Artificial Intelligence and Statistics*, number EPFL-CONF-192374.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

Zhuoye Ding, Qi Zhang, and Xuanjing Huang. 2012. Automatic hashtag recommendation for microblogs using topic-specific translation model. In *COLING (Posters)'12*, pages 265–274.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 593–596. International World Wide Web Conferences Steering Committee.

Maya R Gupta, Samy Bengio, and Jason Weston. 2014. Training highly multiclass classifiers. *Journal of Machine Learning Research*, 15:1–48.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of ACL*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Feng Niu, Benjamin Recht, Christopher Ré, and Stephen J Wright. 2011. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems*, 24:693–701.

Jieying She and Lei Chen. 2014. Tomoha: Topic model-based hashtag recommendation on twitter. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 371–372. International World Wide Web Conferences Steering Committee.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2764–2770. AAAI Press.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.