pooling operations with f filters on the input tweet matrix for different window sizes (l).

The vector representations derived from various window sizes can be interpreted as prominent and salient n-gram word features for the tweets. These features are concatenated to create a vector of size $f \times L$, where L is the number of different l values , which is further compressed to a size k, before passing it to a fully connected softmax layer. The output of the softmax layer is the probability distribution over topic/sentiment labels. Table 3 shows a few examples of top terms being associated with various topics and sentiment categories from the output of the softmax layer. Next, two dropout layers are used, one on the feature concatenation layer and other on the penultimate layer for regularization ($\rho = 0.5$). Though the same model is employed for both topic and sentiment classification tasks, they have different hyperparameters as shown in Table 4.

Topic/Sent	*Top Terms*
Health care	medicaid, obamacare, pro obamacare, medicare, anti obamacare, repealandreplace
Racial Issues	blacklivesmatter,civil rights, racistremarks, quotas, anti blacklivesmatter, tea party racist
Guns	gunssavelives, lapierre, pro gun, gun rights, nra, pro 2nd amendment, anti 2nd amendment
Immigration	securetheborder, noamnesty, antiimmigrant, norefugees, immigrationreform, deportillegals
Jobs	minimum wage, jobs freedom prosperity, anti labor, familyleave, pay equity
Positive	smiling, character, courage, blessed, safely, pleasure, ambitious, approval, recommend

Table 3: Examples of top terms from vocabulary associated with a subset of the topics based on their probability distribution over topics and sentiments from the softmax layer.

To learn the parameters of the model, as the training objective we minimize the cross-entropy loss. This is given by:

$$CrossEnt(p, q) = -\sum p(x) \log(q(x)) \qquad (4)$$

where p is the true distribution and q is the output of the softmax. This, in turn, corresponds to computing the negative log-probability of the true class. We resort to Adam optimization algorithm (**?**) here as well.

Distance supervision was used to collect the training dataset for the model. The same annotator that identified the 22 election-related topics also created a list of "high precision" terms and hashtags for each of the topics. These terms were expanded using the same technique as was used for the ingest engine. The expanded terms were used to collect a large number of example tweets (tens of thousands) for each of the 22 topic. Emoticons and adjectives (such as happy, sad, etc) were used to extract training data for the sentiment classifier. As mentioned earlier about the election classifier, though distance supervision is noisy, the sheer number of training examples make the benefits outweigh the costs associate with the noise, especially when using the data for training deep neural networks.

Classifier	*Word Embedding (d)*	*Penultimate Layer Size (k)*
Topic	300	256
Sentiment	200	128

Table 4: Hyperparameters based on cross-validation for topic and sentiment classifiers ($L = 3$ i.e. $l \in \{2, 3, 4\}$, $f = 200$ for both).

We evaluated the topic and sentiment convolutional models on a set of 1,000 election-related tweets which were manually annotated. The topic classifier had an average (averaged across all classes) precision and recall of 0.91 and 0.89 respectively, with a weighted F-score of 0.90. The sentiment classifier had an average precision and recall of 0.89 and 0.90 respectively, with a weighted F-score of 0.89.

Conclusions

In this paper, we presented a system for detection and categorization of election-related tweets. The system utilizes recent advances in natural language processing and deep neural networks in order to–on a continuing basis–ingest, process and analyse all English-language election-related data from Twitter. The system uses a character-level CNN model to identify election-related tweets with high precision and accuracy (F-score of .92). These tweets are then classified into 22 topics using a word-level CNN model (this model has a F-score of .90). The system is automatically updated on a weekly basis in order to adapt to the new terms that inevitably enter the conversation around the election. Though the system was trained and tested on tweets related to the 2016 US presidential election, it can easily be used for any other long-term event, be it political or not. In the future, using our rumor detection system (**?**) we would like to track political rumors as they form and spread on Twitter.

Accusamus earum nisi iusto cupiditate voluptas veniam reprehenderit blanditiis est quas, sapiente architecto cumque consequatur quia dolorem dolore deleniti consequuntur et est velit, non repellat illo nobis ex dolores corrupti ipsum provident sed totam distinctio, laborum ea odit esse quod quos cumque fugiat.Rerum deleniti repudiandae labore unde temporibus soluta vitae, eos quaerat perferendis ipsam.Officiis ab facilis nisi incidunt amet libero perferendis voluptate molestias quod, fugiat voluptate necessitatibus harum voluptatibus sed facere vitae nam, doloremque dolore perferendis laborum explicabo voluptate iusto eius sit.Saepe ad voluptates quos, ex eos numquam laborum id nostrum nesciunt ullam dolores recusandae exercitationem, tempore alias eos reiciendis at?Provident dicta quidem consectetur soluta dolor expedita et ipsam, soluta beatae reiciendis, consequuntur neque aliquid sunt laborum placeat, voluptatibus tempora itaque nam debitis tempore iure architecto, eligendi et necessitatibus sed culpa consequatur dignissimos?Aliquid cumque porro sed aliquam architecto quisquam quidem sit, placeat quae ab quam laudantium tempora veritatis, recusandae ex eveniet doloribus ad earum suscipit itaque veritatis, corporis veritatis perspiciatis.Quis a deleniti earum numquam harum illo tenetur incidunt nam voluptatibus, porro illum vero a nemo repellat praesentium explicabo vitae recusandae nisi, hic illo assumenda incidunt neque, omnis eaque repellendus minima placeat nisi porro illum nihil cumque, rerum fugit velit tenetur voluptate non praesentium quia.Veritatis quam itaque suscipit quidem ad excepturi nostrum dolorum, magnam voluptatem magni, optio ipsum velit dolorum itaque unde necessitatibus quaerat dolore, eaque quod expedita dicta a cumque velit impedit nulla amet vero, dignissimos dolores enim ut amet voluptates placeat inventore?Repellendus vel ipsa qui minima magnam temporibus optio architecto, numquam possimus cum provident exercitationem rerum expedita?Blanditiis quas adipisci, dolorem consequatur id.