- All URLs are removed. Most URLs are short URLs and located at the end of a message.
- All mentions are converted to a special symbol, for privacy reason. This includes the mentions appearing in a regular message and the user handles at the beginning of a retweet, e.g. ``RT: @bullguy''.
- All cashtags are replaced by a special symbol, to avoid cashtags to gain a polarity value related to a particular time period.
- Numbers following +, – or white space, but not followed by % (e.g. +23.3, +33, -5.52), are converted to a set of special symbols. These symbols reflect the value range of these numbers, and the range of the number determines which symbol it will be converted to. For example, +12.45 => #increase1, +20.22=> #increase2, -21.45=> #decrease2. These numbers are usually about stock price change, and so they bear sentiment information of the message. Different symbols reflect different degrees of price change.
- Similar to the above step, numbers following +, – or white space, and also followed by % (e.g. +23.34%, -5.8%), are also converted to a set of special symbols. These numbers are usually about price or volume changes. But they are based on percentage, which is different from the numbers discussed in previous step. They also convey important sentiment information.

After passing through the above preprocessing steps, the tweets are used to learn the sentiment lexicon and word embedding model.

Phrase Identification: Phrases usually convey more specific meaning than single-term words, and many phrases have a meaning that is not a simple composition of the meanings of its individual words. To identify phrases, we use the approach described in (Mikolov et al. 2013). We first find words that appear frequently together, and infrequently in other contexts. For example, “short sell” is identified as a phrase; while a bigram “they have” is not. By using this approach, we can form many reasonable phrases without greatly increasing the vocabulary size. To identify phrases, a simple data-driven approach is used, where phrases are formed based on the unigram and bigram counts, using this scoring function:

$$Score(w_i, w_j) = \frac{C(w_i, w_j) - \mu}{C(w_i) * C(w_j)} \quad (1)$$

Where $C\ (w_i,\ w_j)$ is the frequency of word w_i *and* w_j appearing together. μ is a discounting coefficient to prevent too many phrases consisting of infrequent words to be generated. The bigrams with score above the chosen threshold are then used as phrases. Then the process is repeated a few passes over the training data with decreasing threshold value, so we can identify longer phrases having several words. For the StockTwits data set, we empirically set the maximum length of a phrase to 4 words in this study. Other parameters are set as the default values used in (Mikolov et al. 2013).

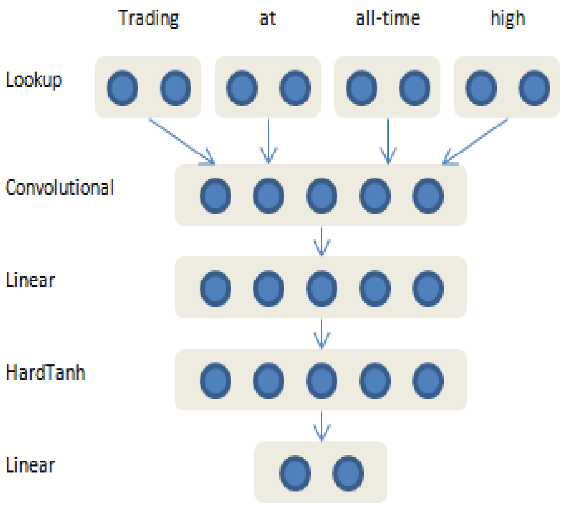

Figure 1: The neural network model for building sentiment lexicon for stock market.

3.2 Sentiment Lexicon Construction

The Proposed Approach: Most corpus-based lexicon construction approaches mainly utilize statistical measures, such as TF-IDF, GI and PMI methods. Our approach is based on a neural network model, inspired by the general network structure for processing NLP tasks (Collobert at al., 2011). Figure 1 shows the neural network we employed for learning the polarity values of a term, by predicting the sentiment value of a StockTwits message. Following (Esuli and Sebastiani, 2006; Vo and Zhang, 2016), we also use two attributes to define the sentiment of a term (word or phrase): positivity and negativity. This means each term has the form of $t = (p, n)$, where p is the positivity value and n is the negativity value. The value range is from 0 to 1 for both p and n. If the value of p is greater than n, we can say that this term has a positive sentiment, and vise versa. If p and n are close to each