the sentiment score for term *t*, using the two TF.IDF values:

$$S_{TF.IDF}(t) = \frac{TF.IDF(t,p) - TF.IDF(t,n)}{TF.IDF(t,p) + TF.IDF(t,n)} \quad (10)$$

The final sentiment class depends on the value of $S_{TF.IDF}(t)$. It is bullish if the value is positive and bearish if it is negative.

PMI is a popular statistic measure used in many previous studies to develop lexicons (Mohammad et al., 2013; Oliveira et al., 2014; Oliveira et al., 2016; Al-Twairesh et al., 2016; Vo and Zhang, 2016). It is defined as:

$$PMI(x,y) = log_2 \frac{p(x,y)}{p(x)p(y)} \quad (11)$$

Where *x* and *y* are two objects, *p(x*) and *p(y)* are the probabilities of occurring *x* and *y* in the corpus, respectively, *p(x, y)* is the probability that they co-occur together. If *x* and *y* are strongly associated, PMI will be largely positive. It is highly negative if they are complementary. And if there is no significant relationship between them, it is near zero. To compute a term's sentiment score, we use both positive and negative PMI values of a term. The S_{PMI} score for term *t* is defined as follow:

$$S_{PMI}(t) = PMI(t, bullish) - PMI(t, bearish) \quad (12)$$

Where *bullish* and *bearish* refer to the sentiment label provided by the message author.

Vo & Zhang approach is a machine learning method that also optimizes the prediction accuracy of message sentiment using lexicons (Vo and Zhang, 2016). To leverage large amount of data, they use a simple neural network to train the lexicon. In this method, each term also has two polarity values: positive and negative. It uses one layer to compute the predicted sentiment probability, by adding the positive and negative values of all the terms in the input message together. Then a softmax function is used to get the predicted sentiment label for the input message. The cross-entropy error is employed as the loss function. Vo and Zhang tested their method on Twitter, using the emotions in a tweet as the indication of its polarity type. They didn't use it in the stock market domain.

3.3 Sentiment-Oriented Word Embedding

Word embedding is a dense, low-dimensional and real-valued vector for a word. The embeddings of a word capture both the syntactic structure and semantics of the word. Traditional bag-of-words and bag-of-n-grams hardly capture the semantics of words (Collobert et al., 2011; Mikolov et al. 2013).

The C&W (Collobert et al., 2011) model is a popular word embedding model. It learns word embeddings based on the syntactic contexts of words. It replaces the center word with a random word and derives a corrupted n-gram. The training objective is that the original n-gram is expected to obtain a higher language model score than the corrupted n-gram. The original and corrupted n-grams are treated as inputs of a feed-forward neural network, respectively. SOWE extends the C&W model by incorporating the sentiment information into the neural network to learn the embeddings (Collobert et al., 2011; Tang et al., 2014b); it captures the sentiment information of messages as well as the syntactic contexts of words. Given an original (or corrupted) n-gram and the polarity of a message as input, it predicts a two-dimensional vector (f_0, f_1), for each input n-gram, where (f_0, f_1) are the language model score and sentiment score of the input n-gram, respectively. There two training objectives: the original n-gram should get a higher language model score than the corrupted n-gram, and the polarity score of the original n-gram should be more aligned to the polarity label of the message than the corrupted one. The loss function is the linear combination of two losses: $loss_0\ (t, t')$ - the syntactic loss and $loss_1\ (t, t')$ - the sentiment loss:

$$loss\ (t, t') = \alpha * loss_0\ (t, t') + (1-\alpha) * loss_1\ (t, t') \quad (13)$$

The SOWE model used in this study was trained from the same 6.4 million StockTwits messages used for building sentiment lexicons; this includes 5.1 million bullish and 1.3 million bearish messages. The metadata of the SOWE model will be presented in the Experiments section

4 Experiments and Results

4.1 Evaluation of Sentiment Lexicons

In this experiment, we evaluated the lexicons built by these approaches: TF.IDF, PMI, Vo & Zhang, and our proposed approach. The same data set, which consists of 6.4 million labeled StockTwits messages, is used by these four methods. The messages are preprocessed accordingly for each method. If the difference between a term's learned positive and negative values is