Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6317–6327
Marseille, 20-25 June 2022

RELATE: Generating a linguistically inspired Knowledge Graph for fine-grained emotion classification

Annika M Schoene[1], **Nina Dethlefs**[2], **Sophia Ananiadou**[1]
[1]National Centre for Text Mining,Department of Computer Science,University of Manchester, UK,
[2] Department of Computer Science,University of Hull,UK
{annika.schoene, sophia.ananiadou}@manchester.ac.uk

Abstract

Several existing resources are available for sentiment analysis (SA) tasks that are used for learning sentiment specific embedding (SSE) representations. These resources are either large, common-sense knowledge graphs (KG) that cover a limited amount of polarities/emotions or they are smaller in size (e.g.: lexicons), which require costly human annotation and cover fine-grained emotions. Therefore using knowledge resources to learn SSE representations is either limited by the low coverage of polarities/emotions or the overall size of a resource. In this paper, we first introduce a new directed KG called 'RELATE', which is built to overcome both the issue of low coverage of emotions and the issue of scalability. RELATE is the first KG of its size to cover Ekman's six basic emotions that are directed towards entities. It is based on linguistic rules to incorporate the benefit of semantics without relying on costly human annotation. The performance of 'RELATE' is evaluated by learning SSE representations using a Graph Convolutional Neural Network (GCN).

Keywords: Sentiment Analysis, Knowledge Graph, Social Media

1. Introduction

A variety of neural networks have been at the core of ground-breaking results in several different research areas in natural language processing (NLP). However, one disadvantage of these models is that they generally require large amounts of labelled data, whilst the knowledge contained in this data could be described in smaller more efficient knowledge representations (Yaqi et al., 2019). Therefore, research efforts have increasingly focused on developing methodologies that make prior knowledge accessible, where one of the most flourishing approaches include embedding representations (Liang et al., 2019). In order to incorporate knowledge into deep learning methods, researchers have focused on building many different resources. There have been three dominant ways to store knowledge in Knowledge-bases (KBs), namely lexicons, ontologies and KGs. Over recent years particularly lexicons and KGs have been successfully created and applied to different NLP tasks. Most notably the creation of lexicons such as WordNet (Miller, 1995) has influenced tasks such as dependency parsing (Herrera et al., 2005). This has also led to the creation of lexicons specific for SA, such as WordNet-Affect (Strapparava et al., 2004) or the NRC emotion lexicon (Mohammad and Turney, 2013). At the same time Language Models (LMs), such as BERT (Devlin et al., 2018) or ELMO (Peters et al., 2019), have been achieving state-of-the-art results in a variety of NLP tasks by incorporating contextual information. However, this approach has often led to words carrying opposing sentiment or emotional meaning having similar vector representations (Zhang et al., 2019), which can impact on worse SA performance (Tang et al., 2015). The key challenge therefore in learning embedding representations that are sensitive towards emotion or sentiment lies in being able to learn word vector representations that not only reflect context but also ensure that emotion words of opposite meanings do not occupy the same vector space. A common approach to overcome this issue relies on using fixed embeddings or fine-tuning them with external resources. These methods range from post-editing already learned embeddings (Yu et al., 2017) to introducing separate 'sentiment channels' to learn new embeddings (Lan et al., 2016). In this paper, we make two contributions to overcome the aforementioned issues in creating fine-grained SSEs. Firstly, a new KG generated from free text is introduced, that contains both fine-grained emotions based on Ekman (Ekman, 1999) and covers a wide range of concepts. Secondly SSE representations are learned using Graph Convolutional Neural Networks (GCNs) to incorporate emotion knowledge implicitly. The SSE representations are compared to existing state-of-the-art LMs in the task of fine-grained emotion classification in tweets. Finally, we present an analysis of the learned representation and outline a number of ethical considerations when utilising social media data to generate embedding representations.

2. Related Work

Existing Knowledge Sources Several KBs have been created with the specific intent to capture human emotions, through both fine-grained emotions and polarities that are associated with words. These often include lexicons, ontologies or KGs, where some of the more advanced KBs also rely on linguistic rules in order to improve the accuracy of the KB. Research conducted by (Cambria et al., 2010) creates a new KB for

opinion mining, where a collection of polarity concepts is created. There have been many iterations of this work leading to the latest release of SenticNet 5 (Cambria et al., 2018), which uses Recurrent Neural Networks to discover concept primitives. OntoSenticNet (Dragoni et al., 2018) was build on top of SenticNet as an ontology for SA tasks. New techniques were developed by (Ofek et al., 2016), that learn polarities of new concepts and therefore increase SenticNet's commonsense affective concepts. Another commonly used resource is the NRC lexicon (Mohammad and Turney, 2013), which is a lexicon created through human annotation using Amazon Mechanical Turk. In this work, around 14,000 words are annotated for both polarities and fine-grained emotions based on the emotion theory proposed by (Plutchik, 1984). The NRC lexicon also contains a suite of different resources that include an emotion hashtag lexicon (Mohammad and Kiritchenko, 2015) and support in different languages (Kiritchenko et al., 2016). Further work by (Mohammad et al., 2013) described the creation of a lexicon based on tweets containing positive and negative emoticons. More recently, work by (Xu et al., 2020) has developed a knowledge graph based on emotion co-occurrence statistics, where each emotion is a node in the graph.

LMs and Sentiment Embeddings Word embeddings can be grouped into *static word embeddings* and *contextualised word embeddings* (Ethayarajh, 2019; Peters et al., 2019). The two most popular static word embeddings are called Word2Vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014). The main disadvantages of these embeddings have been that they cannot take advantage of larger context when producing embeddings representations as these are all based on co-occurrence of words with each other or predicting words from a very short context (skip gram). Therefore, contextualised LMs have gained increasing popularity due to their ability to incorporate context successfully. These models include BERT (Devlin et al., 2018), ELMO (Peters et al., 2018) or ERNIE (Sun et al., 2019). Most of the previously mentioned methods have been applied to SA tasks, but this has often been limited to polarity detection only. Furthermore most work in the space of LM models has focused on finding the semantic and syntactic similarities of words (Bengio et al., 2003), which is natural given the tasks they were originally used for. Less attention has been paid to incorporating emotional context or meaning. Work by (Ren et al., 2016) has argued that one of the limitations of current approaches is that one embedding is generated for each word and it does not take into account that a sentiment-bearing word could be polysemous.

Research conducted by (Liang et al., 2019) proposes the use of refined word embeddings for *Target Aspect Based Sentiment Analysis* (TABSA) that aims to identify aspects in relation to their targets and infer a sentiment from target-aspect pairs. (Yu et al., 2017) proposed a vector refinement model that can be applied to pre-trained word embeddings (e.g., Word2Vec and GloVe), where word embeddings are adjusted so that semantically and sentiment similar words are closer to each other and vice versa. This is done by utilising a sentiment lexicon that contains real-valued sentiment scores. Work by (Tang et al., 2014) introduce Sentiment Specific Word Embeddings (SSWE), where the nearest neighbours in an embedding representation are not only semantically close but also close in sentiment. For this task, sentiment embeddings are learned from tweets using emoticons as labels for positive and negative polarities. (Ren et al., 2016) proposes a method to enrich embeddings with topic and sentiment information to overcome the issue of traditional word embeddings, without taking into account sentiment-bearing words and the context or topic they are used in. Two learning models are introduced to generate Topic Sentiment Word Embeddings (TSWE) and Topic-Enriched Word Embeddings (TEWE). Both models are using an n-gram based neural network (C&W model) that is capable of learning local context and semantic relations. Work by (Maas et al., 2011) uses a probabilistic topic model that derives polarities based on the embeddings of each word. The model is compared to other existing topic models, including LDA and LSA. (Labutov and Lipson, 2013) use logistic regression to re-embed existing embeddings.

Graph Neural Networks Graph Neural Networks (GNNs) were first introduced by (Scarselli et al., 2008) and have since then impacted on a number of different research disciplines such as social science (Kipf and Welling, 2016) and knowledge graphs (Hamaguchi et al., 2017). Since the introduction of the original GNN by (Scarselli et al., 2008), a number of new variations have been developed which include GNNs that use gating and attention mechanisms (Zhou et al., 2018). Most recently work by (Yao et al., 2019) introduced a GCN for text classification, where both word and document embeddings are learned. (Li and Goldwasser, 2019) use GCN to both capture social context in a document and utilise it as a source of distant supervision. Work by (Hu et al., 2021) have utilised GCNs to learn entity embeddings that are integrated into an existing knowledge graph.

3. Emotion Classification Task and Data

The emotion classification task was first introduced by (Klinger et al., 2018), where a Twitter dataset was collected based on emotion keywords that were then removed from the tweet. The aim of the task is then to accurately classify a tweet into an emotion category based on Ekman's six basic emotions. For the benchmarking task the dataset collected by (Klinger et al., 2018) is used. Another dataset was collected based on the same principles (see Figure 1) and then used to create RELATE. The data collection was started in September 2017 and finished in December 2018 and there is an imbalance in the number of tweets acquired per emo-

tion category. This is not intentional and is a product of the availability of data that contains certain keywords in tweets. [1]

Emotion	Tweets	Keywords
Anger	44320	anger,angry, furious
Fear	76718	fear, scared, fearful
Disgust	41742	disgust, disgusting
Surprise	41647	surprise, surprising
Joy	184507	joy, happy
Sadness	48909	sad
Total	398595	

Table 1: Data set for Experiments

3.1. Data Preprocessing

Prior to preprocessing each tweet, the streamed data is checked for any duplicates based on the unique tweet IDs. If any duplicates are found the tweet will be removed. The reason for removing the re-tweets are twofold: (i) collecting re-tweets would introduce duplicates as the original message is included in the stream and (ii) it is hypothesised that re-tweets always involve some form of conversation which might be incomplete due to the way the API works and any emotion keywords might be lost. Also, there are ethical concerns to be taken into account when working with data collected on public social media platforms. This includes but is not limited to identifying information such as a username or mentioned user in a tweet. We consulted the 'Social Media Research: A Guide to Ethics' by (Townsend and Wallace, 2016) to ensure we adhere to all ethical standards. Furthermore, it has to be noted that due to the nature of the data set that the there is an inherent bias, which will be discussed in section 7.1.

4. RELATE

In an effort to include emotions, new KBs have been created either from scratch (Cambria et al., 2010; Mohammad and Turney, 2013) or built on existing resources (Strapparava et al., 2004). However, there seems to be a trade-off between granularity of emotions, where larger KBs only contain polarities (Cambria et al., 2010) and more fine-grained emotion KBs that are smaller in size (Strapparava et al., 2004; Mohammad and Turney, 2013). To overcome this problem between coverage and affective granularity 'RELATE', a KG build on Ekman's six basic emotions is proposed. For this Twitter data introduced in section 3 is utilised. Constructing a KG from natural language is traditionally seen as a challenging task, because of the complex structure of language data (Kertkeidkachorn and Ichise, 2018). A commonly used technique when creating KGs from text is using linguistic theory in the form of semantic parsing (Exner and Nugues, 2012; Carlson et al., 2010; Fader et al., 2011). The following section will outline the preprocessing steps taken to obtain the typical triple structure containing 'Subject-Verb-Object' for the KG.

4.1. Text and Emoji Preprocessing

There are several challenges when working with Twitter data because there is no restriction put upon Twitter users, except the limitation of characters per tweet (maximum of 250 characters per tweet). Therefore people are free to use any form of language in order to communicate their message. This can include colloquialism, well-known acronyms (e.g., BRB = Be Right Back) or emojis (Agarwal et al., 2011) amongst others. Two well-known NLP tools, *Ekphrasis* (Baziotis et al., 2017) and *Spacy* (Explosion, 2017) were used for anonymising and preprocessing the data, which is a common step in producing a new KG (Exner and Nugues, 2012; Cattoni et al., 2012). *Ekphrasis* was used to replace and remove all usernames and URLs with placeholders. This is effective for ensuring any tagged person is not mentioned, however there is still a risk that a person is named by name only (e.g., *'Barack Obama'*). We then decided to replace personal pronouns, such as (*i ' m* to *i am*) and helping verbs to make dependency parsing easier. Similar to (Exner and Nugues, 2012) references, such as mark-ups were removed and only the running text is kept. This also includes the preprocessing of emojis in the data. To get accurate representations of each emoji, *Spacymoji* (Explosion, 2017) was used to identify all emojis. Then the description provided for each entry was used to create a new textual representation (see Figure 1). Overall there were over 1,069 different types of emojis found in this dataset.

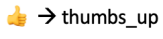

Figure 1: Example of a Emoji to textual description representation

4.2. Sentence Segmentation

To obtain more high-quality triples, sentence segmentation was performed to distinguish between minor and major sentences. Minor sentences usually follow an abnormal pattern and often compromise emotional noises, such as *'ugh!'* or proverbs, e.g., *'easy come, easy go'*, which means that they often do not follow the rules of English grammar (Crystal and McLachlan, 2004). Major sentences mostly follow the rules of English grammar and clauses contain some variation of the Subject-Verb-Object order (Crystal and McLachlan, 2004). Therefore, minor and major sentences are distinguished by splitting each tweet based on three types of punctuation marks: full stop, question and exclamation mark. The authors are aware that this

[1] The authors are happy to share the Tweet IDs for collecting this dataset in accordance with Twitters regulations.

approach does not ensure that all fragments containing less than two words are minor sentence and that there are no minor sentences mistaken for major sentences. Major sentences can be split into two main types—simple and multiple— sentences where simple sentences often contain only one clause and multiple sentences often follow a pattern of *'clause' + 'linking word' + 'clause'* and therefore contain multiple clauses. The four main types of clauses are either simple or linked by coordination or subordination (Crystal and McLachlan, 2004). Three different dictionaries that contain both coordinators and/or subordinators as outlined by (Crystal and McLachlan, 2004) (see Table 2) are generated.

Type of Sentence	Number
Major	617,078
Minor	47,658
Simple	380,079
Compound	86,233
Complex	86,987
Compound Complex	63,779

Table 2: Types of Sentences in Dataset

4.3. Obtaining Triples

Spacy's dependency parser (Explosion, 2017) is used to extract triples from the data. Several different dependency parsers are available for NLP tasks (Loper and Bird, 2002; Chen and Manning, 2014). There are also parsers for Twitter data such as Tweebo (Kong et al., 2014); however, it was found that these were less effective in identifying an appropriate sentence structure. This could be due to the noise and complexity that is present in tweets. Therefore, we decided to split each clause of the type compound, complex and compound-complex based on coordinators as outlined by (Crystal and McLachlan, 2004). This yielded a total of 617,078 clauses before using dependency parsing. Using an approach similar to (Schmitz et al., 2012), all syntactic information given was used to extract the main subject, relation and object from the tweets and annotate each clause with syntactic information.

For clauses containing no coreference, triples were obtained as described in Algorithm 1. We iterated over each clause to find the main *ROOT* (see 4) of the sentence. This is in most instances the main *verb* and it is used as the relation linking the *subject* and the *object*. There is a strict pattern for grammatically correct sentences, where the *subject* is usually found on the left side of the main *verb* and the *object* on the right side. Named Entity Recognition (NER) is often used in these tasks; however, in this instance no existing NER tool was used, because of the lack of coverage existing tools provided. Therefore, both *subject* and *object* are manually identified in each clause by traversing the dependency tree. The *subject* (see 10) was identified to the left of the main *verb*. If there was no left side to the root word, the whole clause was considered to search for the dependency tag *'subj'*. The *object* was identified to the right of the ROOT word (see 10) using the dependency tag *'obj'*. Furthermore, *'modifiers'* and *'compounds'* were added to the identification of the object on the right of the main *verb*. For clauses that contain coreferences, the same methodology as above was used. However, to identify the main subject, Spacy's (Explosion, 2017) neural coreference module was used. For this, the first item in the returned list of coreferences as the main *subject*. Only triples where there is two or more types of each triple identified were kept, e.g.: *Subject* and *Object* or *Subject* and *Relation*. There were 490,299 remaining triples after completing the whole process. There are some downsides to this approach, where we only identify the main triple in each clause and not account for clauses that have more than one *Root*.

input : Single clause with no coreference
output: A triple with the structure Subject-Verb-Object

```
/* Each word in a clause is annotated with a dependency tag      */
relations = []
subjects = []
objects = []
/* Find the main ROOT in each clause and use it as the relation  */
for ( dependency_tag in clause ) {
    if dependency_tag == 'ROOT' then
        relations.append(word)
    end
}
/* Iterate over the subtree to the left of the ROOT and find main
   subject                                                       */
left_root=ROOT.lefts
if len(left_root)==0 then
    dependency_tag == 'SUBJ'
    subjects.append(word)
else
    subject.append(left_root)
end
/* Iterate over the subtree to the right of the ROOT and find main
   object                                                        */
right_root=ROOT.rights
o=""
for ( s in right_root.subtree ) {
    if dependency_tag == 'COMP' then
        o+= word
    end
    if dependency_tag == 'MOD' then
        o+= word
    end
    if dependency_tag == 'OBJ' then
        o+= word
        objects.append(o)
    else
        objects.append("")
    end
}
```

Algorithm 1: Obtaining Triples

5. Learning embedding representations

In the following section, we first outline our approach to learning sentiment-specific embedding representations. Then we describe our experiments comparing the representations against existing approaches.

5.1. Learning model

The learning model used in these experiments to generate embedding representations is a GCN as proposed by (Yao et al., 2019), where both TF-IDF and PMI are used to calculate the edges between nodes in the input KG. More specifically, emotion knowledge is implicitly included in the model by using emotion keywords or triples as nodes (see Figure 2 for an overview). One of the key benefits of using a graph to represent textual

data is, that text is not just seen in as an isolated data point, but as a set of connections containing entities and relations describing the relationships between the texts. The embedding representations are learned with the following experiment setup for the GCN, learning rate = 0.001, hidden units = 200, dropout = 0.5.

Data In order to generate new embedding representations using a GCN, four different datasets are used (see Table 7 in Appendix A). This is done to compare how effective embedding representations based on the knowledge graph 'RELATE' are.

6. Comparison of embedding representations

In the following section, we compare the previously trained embedding representations to popular existing approaches. For this we use the IEST shared task dataset, described in section 3.

6.1. Experiments

A performance baseline is established for the task of classifying the IEST dataset into six different emotion categories, where a simple two-layer LSTM is used with a simple embedding look-up layer. The IEST data is split into 80% training, 10% validation and 10% test data, where the input into each network are 200-dimensional embedding representations unless specified otherwise. All LSTM's share the same hyperparameters, where the learning rate = 0.001, batch size = 128, dropout = 0.5 and the hidden size = 40 units. All experiments were conducted using Tensorflow (Abadi et al., 2016), and early stopping was used to prevent overfitting. The embeddings learned by the GCN are compared against two static word embedding methods GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013a) and two state-of-the-art LMs, including BERT (Devlin et al., 2018) and ELMO (Peters et al., 2019). Each competing embedding method was then used as an embedding layer to the plain vanilla LSTM to see if and how much they contribute to the successful classification of fine-grained emotions in tweets. Whilst fine-tuning was used in both BERT and ELMO; it was decided not to train new LMs based on those architecture using the data provided in section 3, because of the size and nature of these learning models. More specifically, these LMs were developed using large amounts of training data and resources that are nearly impossible to match in an academic setting.

6.2. Results and evaluation

Table 3 show the results for all experiments that were outlined in the previous section, where the top half of the table shows experiments using the GCN, different resources and the bottom half shows the results of the already existing and established embedding models. Firstly, it can be seen that all embedding representations learned by the GCN, except EEK-small, outperform the baseline. Furthermore, it is shown that the best performing embedding representations generated by the GCN are based on the SEMI and RELATE resource. This is especially interesting, because of the size difference and nature of the two resources, which means that the same results can be achieved regardless of whether a GCN is trained on either a larger resource or a linguistically inspired knowledge-graph. Furthermore, it can be seen that the best results were achieved using GloVe, which was pre-trained on 2billion tweets, and the BERT model produced the lowest results. This is particularly surprising, given the amount of data and methodology that is used, and the groundbreaking results that were achieved in other NLP tasks (Gao et al., 2019; **?**). Finally, it can also be seen that ELMO performs better than BERT on this task. This might be due to the 1billion words ELMO was pre-trained on. Table 4 shows the training setup for each model, including the approximate training time (in hours and minutes) and the model parameters for each embedding layer. For all experiments two Tesla P100 GPUs were used. From this it can be inferred that whilst GloVe achieves the best results, 'RELATE' is most efficient to train and can therefore be seen as a more lightweight embedding model. This is also in stark contrast to the time taken by ELMO and BERT to train for 100 epochs. Therefore evidence suggests that (i) it is important which type of data is used to train the embedding model on and (ii) the size of the data is important when not using additional linguistic rule.

Model	Precision	Recall	F-1 Score
LSTM PLAIN	0.52	0.52	0.52
EEK - small	0.47	0.46	0.46
EEK	0.57	0.56	0.56
RELATE	0.58	0.57	0.57
SEMI	0.57	0.57	0.57
Word2Vec	0.52	0.52	0.52
GloVe	0.60	0.59	0.59
ELMO	0.58	0.58	0.58
BERT	0.58	0.59	0.58

Table 3: Experiment results for the GCN embeddings and existing language models using f-1 scores

7. Evaluation

In the following section, we provide two types of analysis for RELATE and discuss ethical considerations.

7.1. Qualitative analysis of RELATE

RELATE is evaluated by firstly looking at the distribution of emotion keywords in each triple. This is done to show how many triples in this directed knowledge graph carry affective meaning. Secondly, in section **??** this KG will be used as a resource to generate new SSE representations.

Distribution of triples Table 5 shows the overall amount of triples and the number of individual entries

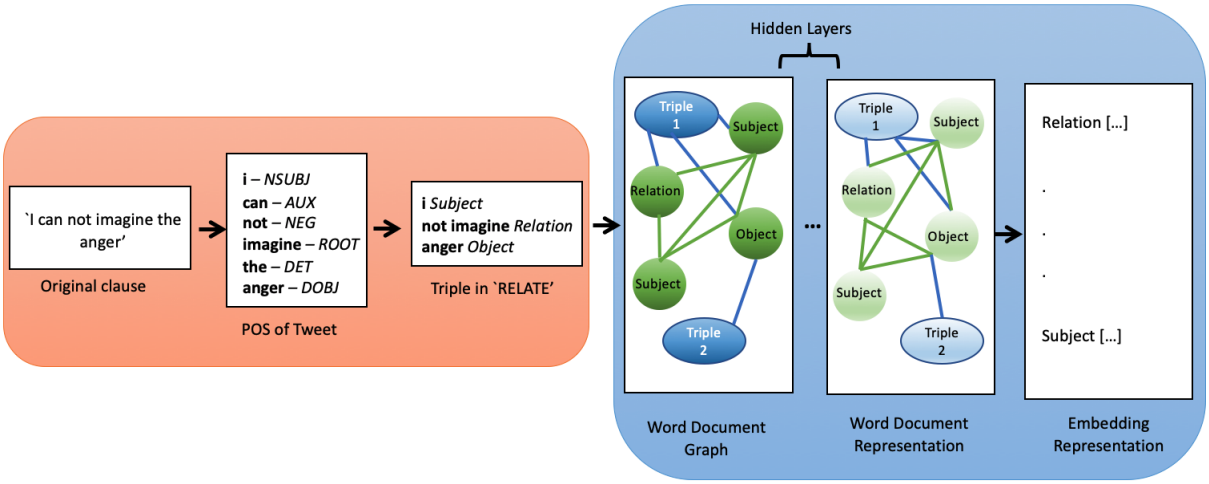

Figure 2: Example of a clause going through preprocessing of 'RELATE' (orange box to the left) and then input into the GCN (blue box to the right - graphic adapted from (Yao et al., 2019)). Blue circles show whole triples and green circles show each individual part of a triple. Green lines indicate word-to-word edges (using PMI) and blue lines show word-to-document edges (using TF-IDF). The first image in the blue box shows the Word-Document graph generated, the second image shows the learned representations for both words and documents and third image shows the output of an m x n dimensional matrix for each triple.

Resource	Comments	Training Time	Embedding Parameters
RELATE		02:46	11,035,400
EEK		04:31	11,035,400
EEK - small		00:36	683,000
SEMI		03:43	11,035,400
Word2Vec		02:43	22,070,800
GloVe		03:11	11,035,400
ELMO		11:38	262,400
BERT	fine_tuning layers = 10	15:18	110,104,890

Table 4: Overview of the different settings, training times (hours :mintues) and model parameters for each embedding layer

for each triple. The total number of triples is *490,299*. Table 6 shows the distribution of emotion keywords in the KG, and it can be seen that 42,646 triples contain emotion keywords, which means that there are around 8.69% of the whole KB contain emotion keywords. It could be argued that when splitting tweets into their sentence segments that there is not necessarily an emotion keyword in each sentence segment. Furthermore, it can be seen that there is a large number of triples in the *happy* emotion category. This is not surprising, because of the initial distribution of keywords in the original dataset.

Discussion A common downside of many KGs is that they only contain the knowledge that was explicitly referenced in the text and therefore fails to capture anything beyond that (Bosselut et al., 2019). This also applies to the emotions represented in this KG, because by default, these are limited to Ekman's six basic emotions. There are many challenges when working with this type of noisy data, which means that many existing NLP toolkits fall short. One such example is using a NER toolkit that is commonly used in KG creation (Mesquita et al., 2019) to detect entities. In our case, we tried existing methodologies; however, this was unsuccessful, where empty values were returned. This is attributed to two main issues: (i) many existing toolkits are trained for texts that are mostly procedural, e.g., news articles or Wikipedia entries, which means

Triple Type	Number of Entries
Subject	400,166
Relation	464,295
Object	323,106

Table 5: Overall number of triples in the knowledge base

Emotion	Subject	Relation	Object	Total
Anger	641	1,143	2,312	4,096
Fear	701	4,019	1,421	6,141
Disgust	484	2,639	1,748	4,871
Surprise	687	3,062	1,062	4,811
Joy	7,632	5,813	3,539	16,984
Sadness	816	2,539	2,388	5,743
Total	10,961	19,064	12,470	42,646

Table 6: Emotion keywords that are part of a triple

that models presented with a different language structure would not work well; and (ii) much of the language used in tweets is non-standard, where previous approaches have often relied on capitalisation to detect entities, and this is often not done in informal language (Mayhew et al., 2019). There are many use cases for an emotion KG in a range of different tasks, such as commercial and health care applications to robotics. This includes tracking the sentiment people express towards a range of different topics (e.g., products or politics), creating dialogue systems that can give an adequate response to emotions expressed by its users or improving Human-Computer-Interaction designs (Mohammad and Turney, 2013).

7.2. Analysis of embedding representations

In order to visualise the embedding representation, Tensorboard Embedding projector (Tensorflow, 2020) was used. This was done for both the best performing GCN representations using RELATE and the representations learned by GloVe. Figures 3 and 4 show the visualisation for the keyword *'joy'*, where the 100 closest words (measured in cosine similarity) are highlighted around the keyword. The x and y - axis are fixed to the left and right for the words 'good' and 'bad' (indicating positive and negative valence) respectively in order to identify bias in the embedding representation. In the following section, we will give two examples of emotion keywords that are categorised as positive valence ('Joy') and negative valence ('sad').

'Joy' emotion keyword It can be seen that for RELATE (3) the emotion word 'joy' is closely associated to concepts of time such as 'winewednesday' or '' for the word 'good'(on the x-axis to the left), but further away from terms such as 'singlesawarenessday' and 'gym' are seen for the word 'bad' (on the y-axis to the right). On the other side, GloVe (4) shows that concepts such as 'love' and 'goodness' are more close to the emotion word 'joy', whereas 'confusion' and 'pain' are further away from it.

'Sad' Emotion Keyword The emotion keyword 'sad' as shown in Figures 5 and 6 is positioned further towards the left in RELATE and the middle in GloVe. Furthermore, it can be seen that words such as 'snapping' and 'superspecial' are close to the emotion keyword. In GloVe words such as 'goodbye' and 'bad' are closer to 'sad'.

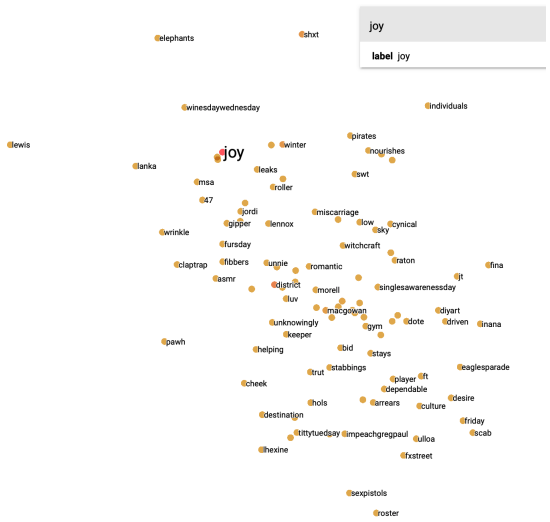

Figure 3: Visualisation of the emotion keyword 'joy' in the embedding representation of the GCN using RELATE

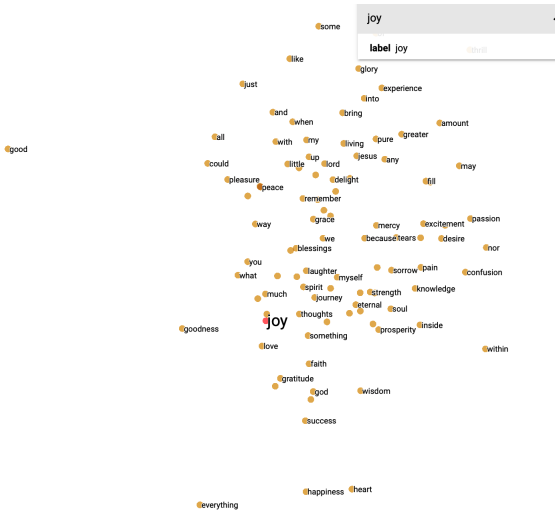

Figure 4: Visualisation of the emotion keyword 'joy' in the embedding representation of GloVe

Discussion This qualitative analysis shows that even though RELATE is far smaller in terms of size, it shows emotion keywords are correctly positioned according to their valence. Furthermore, it can be seen that other words that carry similar emotional meaning (e.g.: 'fine') are also positioned according to the expected valence. It can be seen that GloVe shows more standard/general representations of words and concepts, whilst RELATE includes colloquial and social media language. This also includes hashtags shown in RELATE, but interestingly neither embedding representation shows any preprocessed emoji representations in the 100 closest words as measured by cosine similarity. It can also be seen that often in GloVe words are closely related to their singular and plurals, whereas in RELATE words reflect more current events or thoughts

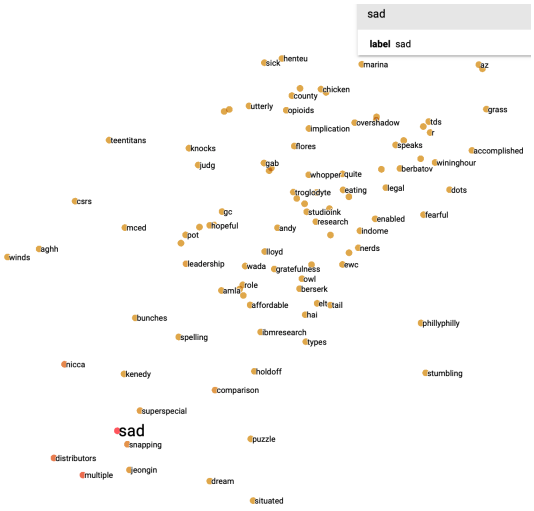

Figure 5: Visualisation of the emotion keyword 'sad' in the embedding representation of the GCN using RELATE.

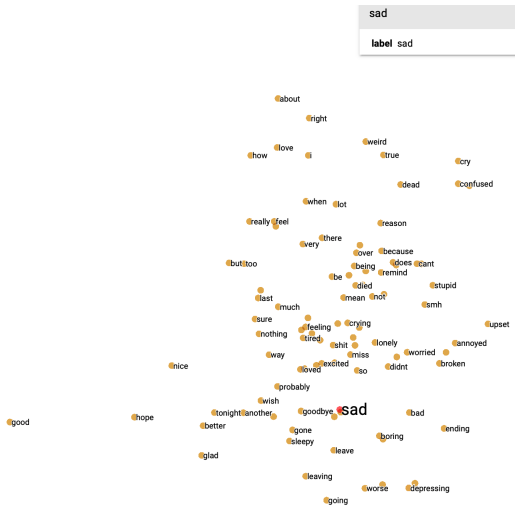

Figure 6: Visualisation of the emotion keyword 'sad' in the embedding representation of GloVe.

and opinions that may have been talked about during the collection of the data (e.g.: The Manchester bombings). Also it was found that 'surprise' is positioned in the middle for both embedding representations and both make references to people. Overall, it could be argued that GloVe is better at capturing general concepts based on the millions of pretraining word that were used in its training, whilst RELATE is showing embeddings that are more closely related to ongoing events at the time of collection and incorporates non-standard language (e.g.: hashtags and colloquial language). On the other hand, RELATE exposes inherit bias in the embedding representation, which if used in real world task could lead to further increasing inequality and discrimination. Therefore, future work should first and foremost look at how to circumvent this bias in embedding representation to create fairer and ethical representations that are not potentially harmful. Furthermore, it should look at pretraining RELATE on a larger dataset to see if these findings change. Also, it should be considered to use not just cosine similarity, but also Euclidean distance to show how words are closely related to each other.

8. Conclusion

In this paper, we first introduced a new resource called 'RELATE', which was then used to train a GCN to generate embedding representations. It was shown that the embeddings generated with the GCN produce similar classification results compared to other LMs. However, one of its advantages is that it is more efficient and requires less resources compared to models such as BERT. Furthermore, it was established that any new LMs need to be tested on a variety of real world tasks, before they are applied to a particular type of task such as Sentiment Analysis. Future work, needs to look at further methods of evaluating this knowledge graph as well as the use of GCNs to learn embedding representations.

9. References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics.

Baziotis, C., Pelekis, N., and Doulkeridis, C. (2017). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., and Choi, Y. (2019). Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Cambria, E., Speer, R., Havasi, C., and Hussain, A. (2010). Senticnet: A publicly available semantic resource for opinion mining. In *2010 AAAI Fall Symposium Series*.

Cambria, E., Poria, S., Hazarika, D., and Kwok, K. (2018). Senticnet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Cattoni, R., Corcoglioniti, F., Girardi, C., Magnini, B., Serafini, L., and Zanoli, R. (2012). The knowledgestore: an entity-based storage system. In *LREC*, pages 3639–3646. Citeseer.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Crystal, D. and McLachlan, E. (2004). *Rediscover grammar*. Longman.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dragoni, M., Poria, S., and Cambria, E. (2018). Ontosenticnet: A commonsense ontology for sentiment analysis. *IEEE Intelligent Systems*, 33(3):77–85.

Ekman, P. (1999). Basic emotions. *Handbook of cognition and emotion*, pages 45–60.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Exner, P. and Nugues, P. (2012). Entity extraction: From unstructured text to dbpedia rdf triples. In *WoLE@ ISWC*, pages 58–69.

Explosion, A. (2017). spacy-industrial-strength natural language processing in python. *URL: https://spacy. io*.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics.

Gao, Z., Feng, A., Song, X., and Wu, X. (2019). Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299.

Hamaguchi, T., Oiwa, H., Shimbo, M., and Matsumoto, Y. (2017). Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. *arXiv preprint arXiv:1706.05674*.

Herrera, J., Penas, A., and Verdejo, F. (2005). Textual entailment recognition based on dependency analysis and wordnet. In *Machine Learning Challenges Workshop*, pages 231–239. Springer.

Hu, L., Zhang, M., Li, S., Shi, J., Shi, C., Yang, C., and Liu, Z. (2021). Text-graph enhanced knowledge graph representation learning. *Frontiers in Artificial Intelligence*, 4.

Kertkeidkachorn, N. and Ichise, R. (2018). An automatic knowledge graph creation framework from natural language text. *IEICE TRANSACTIONS on Information and Systems*, 101(1):90–98.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kiritchenko, S., Mohammad, S. M., and Salameh, M. (2016). Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June.

Klinger, R., De Clercq, O., Mohammad, S. M., and Balahur, A. (2018). Iest: Wassa-2018 implicit emotions shared task. *arXiv preprint arXiv:1809.01083*.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, N. A. (2014). A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012.

Labutov, I. and Lipson, H. (2013). Re-embedding words. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 489–493.

Lan, M., Zhang, Z., Lu, Y., and Wu, J. (2016). Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3172–3179. IEEE.

Li, C. and Goldwasser, D. (2019). Encoding social information with graph convolutional networks forpolitical perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2594–2604.

Liang, B., Du, J., Xu, R., Li, B., and Huang, H. (2019). Context-aware embedding for targeted aspect-based sentiment analysis. *arXiv preprint arXiv:1906.06945*.

Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Mayhew, S., Tsygankova, T., and Roth, D. (2019). ner and pos when nothing is capitalized. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6256–6261, Hong Kong, China, November. Association for Computational Linguistics.

Mesquita, F., Cannaviccio, M., Schmidek, J., Mirza, P., and Barbosa, D. (2019). Knowledgenet: A bench-

mark dataset for knowledge base population. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 749–758.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mohammad, S. M. and Kiritchenko, S. (2015). Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Mohammad, S. M. and Turney, P. D. (2013). Nrc emotion lexicon. *National Research Council, Canada*.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Ofek, N., Poria, S., Rokach, L., Cambria, E., Hussain, A., and Shabtai, A. (2016). Unsupervised commonsense knowledge enrichment for domain-specific sentiment analysis. *Cognitive Computation*, 8(3):467–477.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Peters, M. E., Neumann, M., Logan, I., Robert, L., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219.

Ren, Y., Zhang, Y., Zhang, M., and Ji, D. (2016). Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In *Thirtieth AAAI conference on artificial intelligence*.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al. (2012). Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 523–534. Association for Computational Linguistics.

Strapparava, C., Valitutti, A., et al. (2004). Wordnet affect: an affective extension of wordnet. In *Lrec*, volume 4, pages 1083–1086. Citeseer.

Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., and Wu, H. (2019). Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565.

Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., and Zhou, M. (2015). Sentiment embeddings with applications to sentiment analysis. *IEEE transactions on knowledge and data Engineering*, 28(2):496–509.

Tensorflow. (2020). Tensorboard embedding projector. `https://www.tensorflow.org/tensorboard/tensorboard_projector_plugin`, Oct. Accessed on 2020-10-10.

Townsend, L. and Wallace, C. (2016). Social media research: A guide to ethics. *University of Aberdeen*.

Xu, P., Liu, Z., Winata, G. I., Lin, Z., and Fung, P. (2020). Emograph: Capturing emotion correlations using graph networks. *arXiv preprint arXiv:2008.09378*.

Yao, L., Mao, C., and Luo, Y. (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Yaqi, X., Xu, Z., Meel, K. S., Kankanhalli, M., and Soh, H. (2019). Embedding symbolic knowledge into deep networks. In *Advances in Neural Information Processing Systems*, pages 4235–4245.

Yu, L.-C., Wang, J., Lai, K. R., and Zhang, X. (2017). Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 534–539.

Zhang, X., Wu, J., and Dou, D. (2019). Delta embedding learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3329–3334.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2018). Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*.

A. Appendix A

Resource	Description	Size
RELATE	Triples are used as input into the GCN to generate 200-dimensional embeddings for each triple.	42,646
EEK	The full EEK dataset was used as input to the GCN	390,000
EEK - small	The resources was randomly generated to see how effective RELATE would be when the inputs are plain tweets and the resource size is the same.	42,646
SEMI	Not all triples in RELATE have labels, therefore a GCN was used to automatically label all missing triples with emotion labels so that all knowledge graph triples could be used as input	490,299
Word2Vec	Tweets collected and pre-trained by godin2015multimedia	400 million
GloVe	Tweets collected and pre-trained by pennington2014glove	2billion
ELMO	This LM was taken from Tensorflow-hub `https://tfhub.dev/google/elmo/2`	1Billion
BERT	This LM was taken from Tensorflow-hub `https://tfhub.dev/google/bert_uncased_L-12_H-768_A-12/1`	

Table 7: Overview of the different resources that were used as input to the GCN