Forging the Perfect Recognition

Matthew Aquilina, Aristotelis Anastassiou, Adam DePauw

Abstract

When recognizing or appraising an employee for their work, the aim is to instil a sense of pride and achievement. However, it is often the case that recognitions could be of a poor-quality; failing to achieve its desired impact on the recipient. How would one be able to make such a recognition stronger? How would one rate a recognition's impact in the first place, without requiring a domain-expert's manual appraisal? We propose a deep neural network classifier, trained on a moderated dataset of labelled recognitions, as a solution for quantifying a recognition's impact. Furthermore, we propose a generative approach to predict the most impactful components one could add to a weaker recognition to improve it. Our novel approach has achieved a cross-validated accuracy of 76% when matched against the domain-expert ratings. With our generative approaches, we have successfully shown that the augmented recognitions we produce improve on the rating of poor recognitions.

1. Introduction

Recognition for one's work is one of the major motivational factors for employee satisfaction and retention (Aguenza & Som, 2018) and can take many forms (Brun & Dugas, 2008). Simply providing personal written acknowledgements to employees or even going as far as implementing a formal structured recognition program has been proven to boost productivity and morale company-wide (Bradler et al., 2016; Luthans, 2000). However, for any such program to have a meaningful impact, the actual recognitions themselves need to have value.

In the case of written recognitions (such as a message of appreciation), the quality of the recognition can depend excessively on the writer's skill and is thus highly variable. Being able to quantify a recognition's perceived impact (without significant supervision from a domain expert) would be a useful tool for companies to keep track of their recognition program's efficacy. Furthermore, being able to make suggestions for improving a recognition while the user is writing it should allow for the user to improve the recognition's impact instantaneously, without excessive intervention.

This situation has motivated this project's main research questions. Is it possible to create a classifier capable of rating the *impact factor* of a textual recognition? Taking it a step further, is it feasible to learn to contextualize the structure of an impactful recognition and to generate suggestions for *improving* a recognition?

The classification task shares many traits with some of the most popular sentiment analysis challenges. Long Short-Term Memory Networks (LSTMs) (Gers et al., 1999), deep and/or convolutional neural networks (DNNs/CNNs) and unsupervised encoders (Kiros et al., 2015) have all been used to discern between sentiments in datasets ranging from social media (Mohammad et al., 2018) to video reviews (Maas et al., 2011), sometimes with accuracies as high as 90% (Yenter & Verma, 2017).

On the generative aspect of our task, recent developments claim to achieve human-level parity in text generation in various domains. In particular, OpenAI's GPT-2 (Radford et al., 2019) Transformer-based (Vaswani et al., 2017) architecture is capable of generating short stories of striking authenticity given a prompt (after being trained on millions of webpages). Other architectures using a variety of LSTMs, CNNs and attention mechanisms also produce strong generations while being practical for use with smaller amounts of data (Fan et al., 2018; Radford et al., 2017).

The primary manner in which a recognition classifier (and with extension, a good recognition generator) would differ from the bulk of the sentiment analysis literature is that it would need to measure the intensity and impact (Tian et al., 2018) of a recognition rather than its emotional content or polarity (polarity referring to whether an item contains negative or positive sentiment). Our hypothesis, however, is that sentiment analysis and generator systems can be transferred to this setting if appropriate labelled data indicating the constituents of a strong recognition are provided. This being a niche and previously unexplored area of sentiment analysis, our endeavours in this particular task will be to provide pioneering initial results as an entry-point to this field.

To build up the resources needed to work with one of these classification/generative systems, we procured a domain-expert reviewed dataset of labelled text recognitions from ITA Group's[1] (ITA) internal recognition program (more details in Section 2). From here, we have investigated and built up a number of DNN classifiers utilizing different features of the provided data and various transfer-learning methodologies to correctly classify the impact factor of as many of the recognitions as possible. We provide results

[1] https://www.itagroup.com

which conclusively show that the network has managed to learn the structure of an impactful recognition to a high degree of accuracy.

To tackle suggestion generation, we have trained an LSTM network and the higher-level Skip-Thought vector system (Kiros et al., 2015) to consistently generate recognitions of the highest-rated quality, based on ITA's original dataset. We also present an analysis of how we used these generations to perform suggestions and improve weaker recognitions, with results showing that the generations can indeed be used to improve a weak recognition's impact.

Using the classification and generative models, we present a pipeline for which both classification and suggestion can be incorporated into one system. This allows a user to obtain a real-time review and suggestions for improving a recognition.

The rest of the report is structured as follows. Section 2 expands upon the particulars of the recognition dataset utilized. Section 3 provides a high-level overview of the actual tasks tackled by our classification and generative mechanisms. Section 4 dives into the methodology we followed in exploring classification of the dataset, along with all results obtained. Section 5 follows-up with the process followed for generating plausible high-quality generations from recognitions, with corresponding results. Section 6 details how these generations can be combined with the classification system to suggest back to the user ways to improve a recognition's impact. Finally, Sections 7 and 8 provide scope for future work while placing our own work in context and concluding on the results and achievements obtained in this project.

2. Primary Dataset

Given the nature of the task, specific public datasets for recognition ratings are not available. Thus, we have used a private dataset kindly provided to us from the internal recognition program of ITA Group. ITA specializes in helping organizations implement employee engagement programs, including recognition programs. The dataset contains over 60,000 recognition messages sent to ITA employees by their colleagues over a 10-year period. Each message contains a few sentences of text, with a mean word count of 51 words. 2,390 of the messages were reviewed by domain experts at ITA and given a relative strength indicator of 1-3. A sample recognition (with a 2 rating) from this dataset is provided below (identifying words have been replaced by 'Pnoun'):

"Pnoun, you always work to make sure we are producing a quality product. I appreciate the level of testing you do and the work you have put in."

We applied two types of preprocessing to the data. First, we removed duplicate message texts. Recognitions can be sent to an individual or to an entire team. In the case of the latter, a single recognition message is sent to all members of the team. Each recipient had a single entry in the original dataset. We kept only one copy of each team recognition. After this step the dataset contained 46,486 recognition examples, 2,299 of which were labelled with the strength indicator. We also removed all proper nouns from the recognition messages and replaced them with the token 'Pnoun'. We noted that the names of specific projects, individuals, companies, and places appeared repeatedly in the text, but that these were highly context-specific. While individual proper nouns could be highly useful for a topic modelling exercise, we judged that they would not be useful for our sentiment intensity task. Rather, we determined that any learning from these individual values would not generalize well to future unseen proper nouns. Our generative model, in particular, would make unhelpful recommendations using these proper nouns.

Additional metadata about the sender and receiver of the recognition were also included, such as the team or department, their tenure at the company at the time, and whether the sender and receiver were peers or managers. Further processing of the data was required to engineer features for the classifier, the process of which is described more fully in Section 4.

3. High-Level Task Overview

This section attempts to well-define the two tasks tackled in this project. Both tasks are put into the context of an evaluation pipeline, as shown in Figure 1.

3.1. Classification

Our classification task, as described, is to rate the impact factor of a recognition. To ground our results, we use the metric provided by the domain experts (ITA) to rate recognitions. Specifically, given a text string containing a recognition, the classifier selects a value between 1 and 3 to rate the recognition; 1 being the least impactful and 3 being the most impactful. Thus, the task is not to measure a recognition as containing good or bad sentiment, but simply to measure its strength at delivering its message and affecting the end user.

The end goal is to provide this rating to the writer of the recognition in order for him/her to receive feedback on the quality of the recognition. Thus, this task fits into the lower end of the pipeline in Figure 1 and works to produce the recognition rating at the output side. The evaluation metric used is an accuracy measure of the classifier when averaged from a 10-fold cross validation task on the data provided.

3.2. Generative Suggestions

The generative task's aim is to successfully learn the 'model' of a good recognition and reproduce it consistently, given a primer. The aim of these generations is to always improve upon the seed statement and give it a stronger impact (after classification). However, since recognitions may be context-specific, a general model might not be the best solution for all cases. Thus, further inference could be made to extract

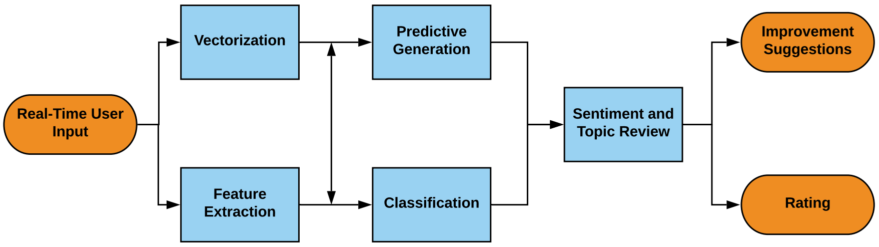


Figure 1. Block diagram showing system pipeline.

the overall topics from these generations and turn them into a format capable of influencing a user to write a better recognition. The generative system takes up the upper end of the pipeline in Figure 1. The evaluation metric used in this case is a direct measure of the effect a generated text has on the impact classification level by concatenating it to the source seed it was generated from and rating this new recognition.

In addition to the classification/generation models, a number of preprocessing mechanisms need to be put in place to turn the text into a format containing information usable by our computational systems. This is especially important since our data is much more scarce than other sentiment analysis datasets, which typically contain hundreds of thousands of labelled examples. These represent the first two blocks in Figure 1 and are explained in more detail in Section 4.

4. Classification and Feature Engineering

This section provides a comprehensive overview of the methodology and results obtained for our classification task. It also provides some insight into the feature engineering carried out on the dataset provided.

4.1. Feature Engineering

Using the dataset, we selected and in some cases further processed data into the feature categories outlined below.

4.1.1. Word Embeddings

Initially, we used `word2vec` embeddings as the input representation of the recognition text, as per Mikolov et al. (2013). The `word2vec` representation for each recognition is simply a weighted sum over the respective vector for each word in the vector space model. We chose this as an initial representation as it is well represented in the Natural Language Processing (NLP) literature.

4.1.2. Recognition Metadata

We included the tenure of both the sender and the receiver measured in number of years within the organization, the team or department name of both the sender and receiver, whether the recognition was sent to an individual or to an entire team, and whether the sender and recipient had either a peer relationship or a supervisory relationship in the organizations hierarchy. We believed these features could be informative and reflect differences in communication styles and semantic features.

4.1.3. Word Counts

We observed that recognition length was highly correlated with higher ratings, so we also included word and character counts of each recognition as an input feature.

4.1.4. Parts of Speech

We extracted parts-of-speech and named entity counters from the texts as a final category of feature inputs. This was motivated by input from our domain experts that highly rated recognitions tend to be more specific about both the exemplary work performed by the recipient, as well as about the positive impact this work had on the sender or organization as a whole. Parts of speech and named entities, then, could serve as a potential measure for this specificity. To engineer these features we used the spaCy[2] open-source NLP library to extract the counts of individual parts of speech as well as named entities that we grouped into three categories: quantitative and ordinal references, date and time references, and proper nouns. The counts were then normalized over the count of all tokens in the text to represent a normalized portion of the text.

4.2. Evaluation

We trained classifiers using all combinations of these feature categories paired with a `doc2vec` vectorization (discussed below). The top performing combinations are shown in Table 1. The results show that counts and the relationship features were the most informative, while the other features did not have a positive impact on performance. Named entity and parts of speech counts likely did not contribute new information because the critical semantic information in these counts was likely already captured by the sentence embeddings. While differences in team culture seemed like an intuitive input, the presence of several dozen teams may mean that the sample size for each team was too low to be informative, given the size of the dataset.

[2] https://www.spacy.io

4.2.1. Transfer Learning and doc2vec

In order to better model our specific data, a `doc2vec` model (Lau & Baldwin, 2016) was implemented as a replacement for the initial `word2vec` model. `doc2vec` directly embeds each individual recognition into a vectorized representation, mapping semantically related text closer together in high-dimensional space. Improved performance over `word2vec` was anticipated, as words which hold little semantic meaning do not have equal weights to more important words.

Because the labelled dataset was relatively small, we explored using a transfer learning approach to improve our classifier's performance. Transfer learning via domain adaptation is an approach that seeks to take information learned from a specific task and reuse it to improve learning on a task from a different domain, on data with a different distribution. In particular, we sought to use a domain adaptation approach that would learn an intermediate representation of text after being trained on a similar, much larger dataset that we could reuse on our own. This has been shown by Glorot et al. (2011) to be effective for semantic analysis tasks. Bengio et al. (2013) provide a good overview of representation learning generally.

We selected two datasets that share some characteristics with our own. The first is a Yelp dataset of service reviews (Yelp, Inc., 2014). This dataset consists of images, reviews, ratings, and categorical information about restaurants, stores, and other services. Our rationale in choosing this dataset is the obvious similarities it shares with our dataset; the reviews are generally conversational, and (in the case of the positive reviews, which we are primarily interested in) offer praise and highlight positive aspects of a given service, which is largely similar to the recognitions in our dataset. The Yelp review dataset consists of over 6.5 million entries, which we believe will improve our model significantly. We trained the `doc2vec` model directly on the Yelp dataset, configured to output 100-dimensional feature vectors.

The second is an Amazon dataset of product reviews that was preprocessed and introduced by Mcauley et al. (2015). This dataset is much larger, with 82 million unique reviews. While the most common ideas included in product reviews differ from those in a recognition for service or exceptional work, the latent semantic features learned from these should be applicable to our task. In particular, the Amazon set has a measure of sentiment intensity inherent in the ratings of the reviews. Because of the large size of this dataset, we used a model that was pre-trained by Radford et al. (2017). The model uses a character-level LSTM model that outputs 4096-dimensional feature representations.

Our doc2vec Model Specifically in our implementation, we train the `doc2vec` model using the following Gensim[3] parameters: `dm=1`, this setting allows for Paragraph Vector Distributed-Bag-of-Words (PV-DBOW) representation of the data, such that single words are learned, but rather sentences are vectorized directly (Lau & Baldwin, 2016); the vector dimensionality is set to 100, for simplicity and quick computation; the learning rate has a maximum of 0.05 which narrows down to 0.01 through training, inspired by the original authors, but with less-drastic values; finally, the model is trained for 20 epochs, the higher end of the recommended Gensim protocol, as the Yelp dataset is considered quite large.

Accuracy	Count	Parts of Speech	Tenure	Team	Relationship
0.7515	✓	✓	✓	✓	✓
0.7510	✓			✓	✓
0.7510	✓		✓	✓	✓
0.7510	✓		✓	✓	✓
0.7500	✓				✓
0.7500	✓	✓			✓
0.7480	✓	✓	✓		✓
0.7480	✓	✓		✓	✓
0.7465	✓		✓		
0.7455	✓			✓	

Table 1. Top 5 performing feature sets, trained on baseline architecture with `doc2vec` word embeddings.

4.3. Network

We performed an initial architecture search using a baseline input of the `word2vec` feature representations. These networks were trained using between 1 and 6 hidden layers, each using 20, 50, and 80 hidden ReLU units, with a three-class softmax output. The system was trained in mini-batches of 100, using the Adam regularization algorithm (Kingma & Ba, 2014) with a learning rate of 1×10^{-5}. Each network was trained for 30 epochs using a cross-entropy loss calculation. Networks were evaluated using 10-fold cross-validation to maximize the number of training examples, with an average of the best validation accuracy for each fold. Initial experiments with varying numbers of layers and hidden units showed the best performing architecture was a 5-layer architecture with 80 hidden units, achieving an initial validation accuracy of 68%.

After further feature engineering experiments, we did a follow-up architecture search to validate with the new feature set. We found we could achieve nearly optimal performance with a simpler 2-layer, 20-hidden unit architecture, and so opted for this simpler design.

Oversampling There was a slight imbalance in our dataset. 50% of the recognitions were rated 2, 30% as 3, and 20% as 1. While this imbalance seems likely representative of the real-world distribution of such recognitions, we wanted to test whether this imbalance had a negative impact. To this end we implemented oversampling using both the ADASYN (He et al., 2008) and SMOTE (Chawla et al., 2002) algorithms. SMOTE works by generating new instances of the minority class by interpolating between minority examples and a number of their k nearest neighbors. For our purposes, we used SMOTE with a value of $k = 5$. ADASYN works similar, but generates new instances close to the minority examples that are nearer to majority class

[3] https://radimrehurek.com/gensim/

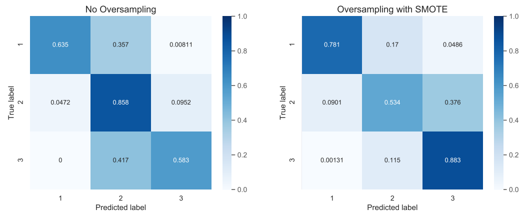


Figure 2. Confusion matrix of classification accuracy without oversampling (left) and with (right).

instances and thus more difficult to learn.

Both approaches reduced our cross-validation accuracy slightly, with SMOTE (0.7365) outperforming ADASYN (0.714) slightly. While oversampling did not increase overall classification accuracy, it did have a noticeable increase in accuracy of the majority classes, with the SVM variant of SMOTE performing best. In particular, it reduced the number of times the rating is underestimated. See Figure 2 for details. We note that this may be a desirable outcome for the target recommendation system, as the user experience cost of failing to recommend changes to a partly flawed recognition is likely to be lower than mistakenly suggesting changes to an already excellent recognition.

5. Generation

This section delves into the process followed to produce a generator capable of producing recognitions of a high quality. We propose two different architectures based on LSTMs and Skip-Thoughts; their results and comparisons are also presented in this section.

5.1. LSTM

Our first generation attempt was using a simple LSTM text generation at a character level (Gers et al., 1999). This model generates a user-specified number of characters given a previous sequence of characters (such as a sentence from a recognition). When training, we fed the LSTM all 3-classified recognitions in sequences of 50 characters with a batch size of 50. Two total LSTM layers (stacked) were used with a hidden dimension of 512. Training was carried out using the Adam optimizer with a weight decay coefficient of 1×10^{-5} and a dropout value of 0.4 over 50 epochs.

5.2. Skip-Thoughts

The Skip-Thought model is a form of sequence-to-sequence model (Kiros et al., 2015). Given a sequential ordered set of sentences, $S = (s_{t-1}, s_t, s_{t+1})$, the Skip-Thought model is trained to predict the previous/next sentences s_{t-1} and s_{t+1} for each source sentence s_t. This is a very good fit for our task since predicting neighbouring sentences will have semantic connections and value to the sentence in question, s_t.

The architecture of the Skip-Thought model is largely ubiquitous in modern NLP approaches. It uses an encoder-decoder setup, employing Recurrent Neural Networks (RNNs) for the encoding and decoding steps. The Skip-Thought model is dependent on contiguous data, which was not exactly in line with our resources. Our recognitions consist of small chunks of contiguous data, and alongside its exceptionally high computational cost (several weeks to compute a single epoch on a high-end GPU), these factors make training the model ourselves unfeasible. We used a pretrained model from the original authors as an encoder instead, and only trained a decoder specifically for our task. This encoder was trained on a corpus of novels, spanning different genres which the authors claim will ensure the model is not particularly biased to a given domain.

The model uses Gated Recurrent Units (GRUs) (Cho et al., 2014) as the recurrent architecture in both the single-layer encoder and single-layer decoder. GRU units offer improvements over traditional RNNs and competitive results with LSTMs (Chung et al., 2014). The vector dimensionality was set to 2400, with the encoder trained for 5 epochs over several weeks. In order to train the decoder for our task, we fed the decoder pairs of sentences from our recognition dataset. All pairs of sentences fed into the system were contiguous. The results were trained over 5 epochs using a GPU to speed up training to a feasible few hours.

5.3. Results

Our LSTM and Skip-Thought models were both evaluated for their suitability in our generate-and-recommend task. We hypothesized that our Skip-Thoughts implementation would significantly outperform the LSTM. Since Skip-Thoughts are trained to predict previous and next sentences in a sequence, we expected it to better capture the semantic intention of an author, as opposed to the character-generating LSTM.

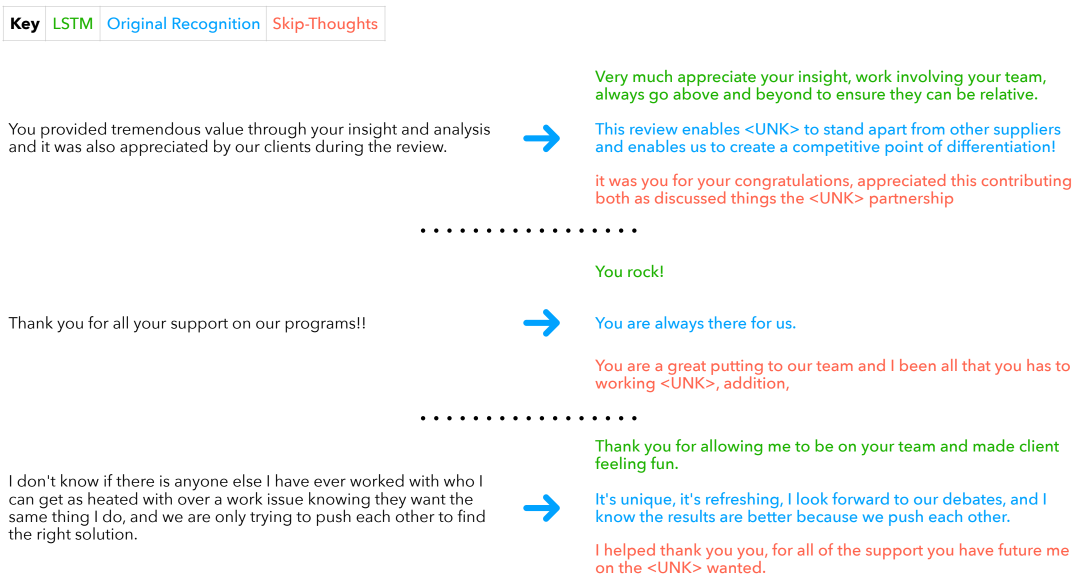
Figure 3 shows samples from our generated sentences. At first glance, the LSTM appears to have learned better grammar, with fewer errors in general. The Skip-Thought model seems to generate key words in a less-grammatical but potentially more semantically-related fashion.

In order to quantitatively evaluate the effectiveness of each of these generative approaches, we pursued two primary questions:

(a) Do these models generate recognitions that increase semantic quality?

(b) Do the generated sentences follow the semantic content intended by a given author?

In order to address (a), we made use of our classifier (Section 4) to predict the rating of an artificial, generated recognition and compare it against the genuine recognitions. In order to make these comparisons fair, we used a test set of recognitions which our generative models had not previously seen (these included recognitions labelled by domain



Key	LSTM	Original Recognition	Skip-Thoughts

You provided tremendous value through your insight and analysis and it was also appreciated by our clients during the review.

Very much appreciate your insight, work involving your team, always go above and beyond to ensure they can be relative.

This review enables <UNK> to stand apart from other suppliers and enables us to create a competitive point of differentiation!

it was you for your congratulations, appreciated this contributing both as discussed things the <UNK> partnership

Thank you for all your support on our programs!!

You rock!

You are always there for us.

You are a great putting to our team and I been all that you has to working <UNK>, addition,

I don't know if there is anyone else I have ever worked with who I can get as heated with over a work issue knowing they want the same thing I do, and we are only trying to push each other to find the right solution.

Thank you for allowing me to be on your team and made client feeling fun.

It's unique, it's refreshing, I look forward to our debates, and I know the results are better because we push each other.

I helped thank you you, for all of the support you have future me on the <UNK> wanted.

Figure 3. Sample seed sentences from recognitions with follow-up ground-truth sentences and generated text samples from the LSTM and Skip-Thought models.

experts as well as by our classifier). We generated the artificial recognitions by using the first sentence of each test set sample to feed into each respective generative model, and create artificial recognitions with the same number of sentences as the original recognition for the Skip-Thoughts and the same number of characters for the LSTM. This made sure no bias from recognition length could affect the classifier.

The genuine recognitions alongside the generated counterparts are grouped by the genuine recognitions' initial classification such that improvements per class can be interpreted and are presented in Figure 4. Results across both models are largely positive. Recognitions that were classified as 1s and 2s show improvements when compared to the artificial recognitions. Recognitions that were already 3s showed no improvement, and actually are scored worse. But this is expected as 3-rated recognitions are impossible to improve. These recognitions still maintain a high-rating nevertheless. Interestingly, in this experiment, the LSTM seemed on average to improve recognitions slightly more than our Skip-Thought model, which went against our hypothesis that Skip-Thoughts should outperform a simple character-generating LSTM.

To address (b), we designed a semantic similarity comparison method within the context of our test set samples. Similar to (a), we use the first sentence of each sample as a seed for the generation of an artificial recognition, except in this setup we limit the generation to a single sentence. We then take this generated sentence per recognition, and compare its similarity to the second sentence of the genuine recognition. This similarity was implemented using our `doc2vec` model (see 4.2.1) for vectorization and a standard cosine similarity score. In order to determine if the `doc2vec` model was representing our specific dataset well

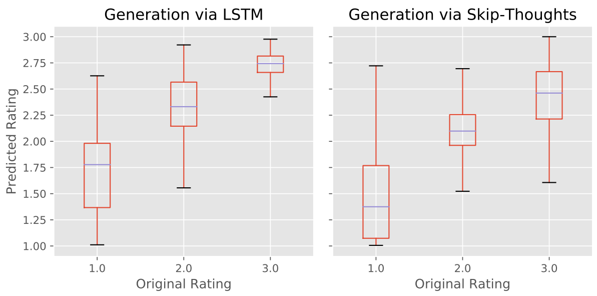


Figure 4. Boxplot representations of the classification differences in generated samples of both the LSTM and Skip-Thought models. Outlier values are disregarded.

enough, we sampled a few sentences from the dataset, and hand-engineered paraphrased samples. One such example is shown below.

Original I am so impressed with your attention to detail and the way you are able to manage several projects so flawlessly.

Paraphrase Your attention to detail is impressive and you are a good manager of many projects.

Empirically, paraphrased samples scored a cosine similarity of over 0.75, nearing 0.8, while randomly sampled pairs scored in general less than 0.3. By computing the similarity for all of these pairs, we calculated the average similarity score for the LSTM model to be 0.35, while the Skip-Thoughts had an average similarity of 0.46. While these do not show exceptionally high similarity scores an ideal model would produce, they do suggest and reinforce our initial experimental hypothesis that Skip-Thoughts learn better semantic relationships in the sequences of sentences than the LSTM.

6. Recognition Improvement

With feasible generators in place, we looked to move from predictions to recommendations for improvement by looking for topics to extract from the data. We used Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to attempt to cluster the data, and then manually reviewed term frequencies to identify common topics or themes within these clusters. We found that cluster separation was strongest with 3 clusters, however these clusters did not correlate with recognition ratings, and the topical terms within them strongly overlapped.

We then asked the domain experts at ITA to provide us with what they believe best contributes to a highly rated recognition. Rather than topics, they provided us with textual qualities that could be present in texts with a wide topical range. We believed, however, that they could still be identified in the latent semantic representations. In addition to defining these qualities, the domain experts also provided us with ideal representations of these qualities, both artificially created and mined from the existing recognition data, which we split into 4 sets of representative sentences we called quality examples. The qualities are as follows: A good recognition should be personal, addressing the recipient directly. It should be timely, not issued months after the exemplary work was completed. It should be connected to the specific task or accomplishment the recipient completed and it should be specific about the value or benefit of that work to the sender and to the organization as a whole.

The process we propose for identifying and recommending one of these qualities to an author is as follows: As a sender writes, we classify the recognition text. If the text falls short of the highest rating, we feed the author's work into the generative model and predict what the next sentence of a highly-rated recognition would be. We then estimate which of the quality examples this sentence is closest to, using a cosine similarity. Finally, we suggest the quality best represented by the predicted sentence to the author as a possible improvement to their work.

The distribution of quality example similarities shows a desirable correlation to the class labels. This increased our confidence a similarity approach could succeed. An example can be seen in Figure 5.

To quantitatively evaluate the quality of these recommendations, we developed an approximation of an author's application of the quality suggested. Deploying this recommender in a user trial would be a more thorough extension of this evaluation, and a more qualitative evaluation could be performed by comparing the recommendations with those of domain experts reviewing individual texts in a blind trial. We leave these more detailed options to future work.

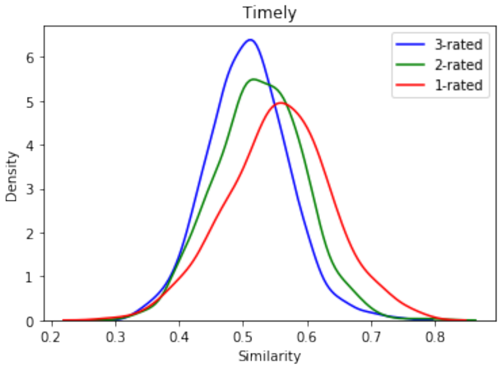


Figure 5. Distribution of similarities to the Timely quality attribute for each rating of the labelled examples. 0 is most similar.

	LSTM	SkipThoughts	Random
No change	2.06	2.06	2.06
1 sentence replacement	2.26	2.26	2.15
2 sentence replacement	2.34	2.36	2.16

Table 2. Expected value of class probabilities before and after sentence replacement.

To approximate user implementations of the recommendations we first divided the entire dataset into over 152,000 individual sentences, encoded the sentences using our `word2vec` word embeddings, and then measured each sentence's cosine similarity to the quality example sentences. We then generated quality recommendations using both the LSTM and the SkipThoughts generative models for each of 2300 unlabelled recognitions. The recognitions were held out from the training of our generative systems. For each recognition, we took the quality recommendation and randomly selected a sentence from the full corpus that was among those most similar to the quality recommended. We then replaced the last sentence of the recognition with the sentence exemplifying the quality, and remeasured the softmax class probabilities. We then repeated this, replacing two sentences instead of one. As a control, we performed the same replacement experiments with sentences chosen completely at random, without consideration of the recommended qualities. In each case, we were careful to replace rather than add on sentences, to ensure we limited any increase in recognition length.

The results in Table 2 show a consistent increase in the expected value of the class probability compared to random sentence replacement, with the best performing model improving the expected value by 0.2 over random replacement. Further, Figure 6 shows that the shift in the probability distributions of the best and worst ratings is very noticeable. The effect of qualities recommended by the LSTM and Skip-Thoughts appear nearly identical, with Skip-Thoughts slightly outperforming the LSTM.

7. Related Work

Our endeavours have touched on two popular subdomains of research in NLP: sentiment analysis and text prediction

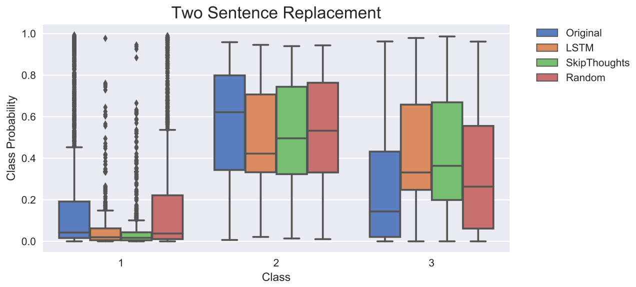


Figure 6. Boxplot representations of the class probabilities of original recognitions, recognitions modified with replacement sentences with qualities recommended using LSTM and Skip-Thoughts, and those replaced with random sentences.

(Hirschberg & Manning, 2015). There are several other techniques which could work within our context and possibly improve upon our results.

7.1. Word Embeddings with DictRep

DictRep is an approach to learning distributed representations of words based on dictionary definitions (Hill et al., 2016a;b). These authors design a system that uses common dictionary definitions of a given concept to map to this concept. This approach leverages lexical and phrase-based semantics to generate high-dimensional data, and could have been used in place of our `doc2vec` model. The advantages of such a model could offer significant benefits to our system; we could use this dictionary-based model to define define topics aided by domain-experts. This would essentially allow for semi-supervised topic discovery in a primarily unsupervised setting.

7.2. FastSent

FastSent is a similar architecture to Skip-Thoughts, but is designed not to be computationally as expensive (Hill et al., 2016a; Polajnar et al., 2015). Instead of using dense representations for sentences individually, FastSent uses a sum across word representations. Such a model could have been useful in our system as we did not aim to generate human-readable text, but rather text from which a topic could be gleaned. Word order, in other words, would not be maintained in such a system.

7.3. Sentiment Analysis using Limited Data

In our methodology, we chose to use transfer learning to extract the knowledge from other pre-trained models when classifying recognitions; thus applying a general method to our specific niche in sentiment analysis. However, other techniques exist for transferring information between domains.

Meta-learning (Vilalta & Drissi, 2002), is an approach widely used to build models capable of learning from experiences on previous tasks. Their particular advantage is that they can learn very quickly from limited data (a process known as few-shot learning) by using their previous experience to quickly adapt to a new task. In sentiment analysis, meta-learning has been applied to tasks under specific frameworks dedicated for the task in question. Earlier approaches used architectures such as Support Vector Machines (SVMs) to introduce meta-learning in NLP (Morante & Daelemans, 2009). More recent methods introduced neural networks (such as the Multitask Question Answering Network (McCann et al., 2018)) and Bayesian frameworks such as Lifelong Sentiment Classification (Chen et al., 2015), to learn from related problems and generalize to all possible tasks directly. The latest few-shot learning approaches (Yu et al., 2018) utilize CNN architectures and adaptive metrics to perform well in a diverse range of few-shot tasks.

Additionally, advances in deep neural networks used in other domains have led to architectures such as Model Agnostic Meta Learning (MAML) (Finn et al., 2017) which are generalized frameworks capable of meta-learning using any model in any domain. From here, generalized systems (such as Antoniou et al. (2018)) have emerged which could theoretically be modified to learn on any sort of task.

7.4. NMT Models with Attention

Skip-Thought models seem to be less-popular in recent literature, with Neural Machine Translation (NMT) models becoming much more mainstream. Specifically, the introduction of attention mechanisms (Vaswani et al., 2017) has proven to be of great value to many different systems, primarily in translation tasks (Junczys-Dowmunt et al., 2016; Sennrich et al., 2017). Adapting this to work for language generation in the space of sequential data, as we had in our dataset, could be an interesting development. Attention particularly may be of as a form of explanatory analysis to define why certain topics or phrases are recommended based on what has already been written.

Future improvements on our system could introduce one or more of these mechanisms mentioned here, in an effort to squeeze out as much information as possible from the limited recognition set and improve classification accuracy.

8. Conclusions

We have built a DNN classifier that can predict the rating of a recognition with 76% cross-validated accuracy against a domain-expert's rating. We have also demonstrated that it is possible to generate recommendations that will improve the quality of recognitions in a measurable way. Feature engineering and tests have shown that impactful recognitions indeed do share some common traits which could be exploited in the future for better classifications and suggestions. Further validation would be needed to demonstrate the feasibility of this approach with real authors making recognitions and receiving suggestions. Many other approaches are available for modifying and improving our system; this work presents a novel baseline to which future work could compare to.

References

Aguenza, Benjamin Balbuena and Som, Ahmad Puad Mat. Motivational factors of employee retention and engagement in organizations. *IJAME*, 2018.

Antoniou, Antreas, Edwards, Harrison, and Storkey, Amos. How to train your MAML. *arXiv preprint arXiv:1810.09502*, 2018.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.50.

Blei, David M, Ng, Andrew Y, and Jordan, Michael I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Bradler, Christiane, Dur, Robert, Neckermann, Susanne, and Non, Arjan. Employee recognition and performance: A field experiment. *Management Science*, 62(11):3085–3099, 2016.

Brun, Jean-Pierre and Dugas, Ninon. An analysis of employee recognition: Perspectives on human resources practices. *The International Journal of Human Resource Management*, 19(4):716–730, 2008.

Chawla, Nitesh V, Bowyer, Kevin W, Hall, Lawrence O, and Kegelmeyer, W Philip. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Chen, Zhiyuan, Ma, Nianzu, and Liu, Bing. Lifelong learning for sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pp. 750–756, 2015.

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Fan, Angela, Lewis, Mike, and Dauphin, Yann. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 889–898, 2018.

Finn, Chelsea, Abbeel, Pieter, and Levine, Sergey. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Gers, Felix A, Schmidhuber, Jürgen, and Cummins, Fred. Learning to forget: Continual prediction with lstm. 1999.

Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 513–520, 2011.

He, Haibo, Bai, Yang, Garcia, Edwardo A, and Li, Shutao. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328. IEEE, 2008. ISBN 978-1-4244-1820-6. doi: 10.1109/IJCNN.2008.4633969. URL https://sci2s.ugr.es/keel/pdf/algorithm/congreso/2008-He-ieee.pdf.

Hill, Felix, Cho, Kyunghyun, and Korhonen, Anna. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*, 2016a.

Hill, Felix, Cho, Kyunghyun, Korhonen, Anna, and Bengio, Yoshua. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016b.

Hirschberg, Julia and Manning, Christopher D. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

Junczys-Dowmunt, Marcin, Dwojak, Tomasz, and Sennrich, Rico. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. *arXiv preprint arXiv:1605.04809*, 2016.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Kiros, Ryan, Zhu, Yukun, Salakhutdinov, Ruslan R, Zemel, Richard, Urtasun, Raquel, Torralba, Antonio, and Fidler, Sanja. Skip-thought vectors. In *Advances in neural information processing systems*, pp. 3294–3302, 2015.

Lau, Jey Han and Baldwin, Timothy. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

Luthans, Kyle. Recognition: A powerful, but often overlooked, leadership tool to improve employee performance. *Journal of Leadership Studies*, 7(1):31–39, 2000.

Maas, Andrew L, Daly, Raymond E, Pham, Peter T, Huang, Dan, Ng, Andrew Y, and Potts, Christopher. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150. Association for Computational Linguistics, 2011.

Mcauley, Julian, Pandey, Rahul, and Leskovec, Jure. Inferring Networks of Substitutable and Complementary Products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 785–794, 2015. ISBN 978-1-4503-3664-2. URL http://dx.doi.org/10.1145/2783258.2783381.

McCann, Bryan, Keskar, Nitish Shirish, Xiong, Caiming, and Socher, Richard. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 3111–3119, 2013.

Mohammad, Saif, Bravo-Marquez, Felipe, Salameh, Mohammad, and Kiritchenko, Svetlana. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 1–17, 2018.

Morante, Roser and Daelemans, Walter. A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pp. 21–29. Association for Computational Linguistics, 2009.

Polajnar, Tamara, Rimell, Laura, and Clark, Stephen. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pp. 1–11, 2015.

Radford, Alec, Jozefowicz, Rafal, and Sutskever, Ilya. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Radford, Alec, Wu, Jeff, Child, Rewon, Luan, David, Amodei, Dario, and Sutskever, Ilya. Language models are unsupervised multitask learners. 2019.

Sennrich, Rico, Firat, Orhan, Cho, Kyunghyun, Birch, Alexandra, Haddow, Barry, Hitschler, Julian, Junczys-Dowmunt, Marcin, Läubli, Samuel, Barone, Antonio Valerio Miceli, Mokry, Jozef, et al. Nematus: a toolkit for neural machine translation. *arXiv preprint arXiv:1703.04357*, 2017.

Tian, Leimin, Lai, Catherine, and Moore, Johanna D. Polarity and Intensity: the Two Aspects of Sentiment Analysis. 7 2018. URL http://arxiv.org/abs/1807.01466.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Vilalta, Ricardo and Drissi, Youssef. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Yelp, Inc. Yelp Dataset, 2014. https://www.yelp.com/dataset, Last accessed on 2019-02-07.

Yenter, Alec and Verma, Abhishek. Deep CNN-LSTM with combined kernels from multiple branches for IMDB Review Sentiment Analysis. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 540–546. IEEE, 2017.

Yu, Mo, Guo, Xiaoxiao, Yi, Jinfeng, Chang, Shiyu, Potdar, Saloni, Cheng, Yu, Tesauro, Gerald, Wang, Haoyu, and Zhou, Bowen. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*, 2018.