Machine Floriography: Sentiment-inspired Flower Predictions over Gated Recurrent Neural Networks

Avi Bleiweiss
BShalem Research, Sunnyvale, U.S.A.

Keywords: Language of Flowers, Gated Recurrent Neural Networks, Machine Translation, Softmax Regression.

Abstract: The design of a flower bouquet often comprises a manual step of plant selection that follows an artistic style arrangement. Floral choices for a collection are typically founded on visual aesthetic principles that include shape, line, and color of petals. In this paper, we propose a novel framework that instead classifies sentences that describe sentiments and emotions typically conveyed by flowers, and predicts the bouquet content implicitly. Our work exploits the figurative Language of Flowers that formalizes an expandable list of translation records, each mapping a short-text sentiment sequence to a unique flower type we identify with the bouquet center-of-interest. Records are represented as word embeddings we feed into a gated recurrent neural-network, and a discriminative decoder follows to maximize the score of the lead flower and rank complementary flower types based on their posterior probabilities. Already normalized, these scores directly shape the mix weights in the final arrangement and support our intuition of a naturally formed bouquet. Our quantitative evaluation reviews both stand-alone and baseline comparative results.

1 INTRODUCTION

Communicating through the use of flower meanings to express emotions, also known as Floriography, has been a traditional and well established practice around the world for centuries. Nonetheless, endorsing this coded exchange as a Language of Flowers only gained traction during the Victorian era, and was backed by publishing a growing compilation body of floral dictionaries that explained the meanings of flowers. Shortly thereafter, the language of flowers was favored as the prime medium to send secret messages otherwise prohibited in public conversations. Floriography was not only about the simple emotion attached to an individual flower, but rather what portrayed in a combination of petals and thrones placed in an arranged bouquet. The language had since developed considerably, and today several online resources (Roof and Roof, 2010; Diffenbaugh, 2011) provide updated flower interpretations and most authentic sentiment transcriptions.

In this research, we investigate the linguistic properties of the language of flowers using unsupervised learning of word vector representations (Mikolov et al., 2013a; Pennington et al., 2014), and modeling the language after neural machine translation that predicts a definitive flower type given a sentiment phrase as input. Furthermore, we extend the single target perspective of the language and relate the short-text sentiment sequence to a plurality of flowers that combine both a principal or pivotal flower, with statistically ranked subordinate flowers to form a bouquet.

Recurrent Neural Networks (RNN) recently became a widespread tool for language modeling tasks (Sutskever et al., 2014; Hoang et al., 2016; Tran et al., 2016). In our case study, we feed sequentially concatenated translation records into a shallow RNN architecture that consists of an input, hidden, and output layers (Elman, 1990; Mikolov et al., 2010). At every time step, the output probability distribution over the entire language vocabulary renders our framework for an automatic selection of sentiment-aware flower species that requires minimal human counseling. We ran experiments on both a standard RNN that applies a hyperbolic activation function directly and through a gated recurrent unit (GRU) (Cho et al., 2014; Chung et al., 2014), and confirmed GRU to better sustain vanishing propagating gradient (Hochreiter and Schmidhuber, 1997) and improve our recall performance.

The main contribution of our work is an effective neural translation model we apply to a small corpus comprised of extremely short-text sequences, by sharing representation power of context both adja-

Bleiweiss, A.
Machine Floriography: Sentiment-inspired Flower Predictions over Gated Recurrent Neural Networks.
DOI: 10.5220/0006583204130421
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 413-421
ISBN: 978-989-758-275-2

cent in time and closely related in semantic vector space. The rest of this paper is organized as follows. In Section 2, we give a brief review of our compiled version of the Language of Flowers and the use of Word2Vec word embeddings to represent sentiment sentences and flower names. Section 3 then overviews the gated recurrent unit (GRU) extension to a standard RNN, and derives our neural network architecture for predicting bouquet flower candidacy from a sentiment phrase. As Section 4 motivates the order of feeding RNN semantically-close sentiment vectors to improve accuracy. We proceed to present our methodology for evaluating system performance end-to-end, and report extensive quantitative results over a range of experiments, in Section 5. Summary and identified prospective avenues for future work are provided in Section 6.

2 LANGUAGE OF FLOWERS

The online floral dictionaries we obtained sort flowers by name and distribute respective meanings in instructive alphabetical chapters (Figure 1). Internally, we represent the language of flowers in a size-l named-member list of variable-length sentiment phrases, each identified with a single pivotal flower f, as illustrated in Table 1. To keep the plant names uniquely labeled in our implementation, those composed of multiple words make up a hyphenated compound modifier. In total, we tokenize and lowercase $l = 701$ sentiment-flower pairs of distinct flower types, as we pare down 3,857 unfiltered words to build a succinct vocabulary of 1,386 symbols, after removing stop words and punctuation marks.

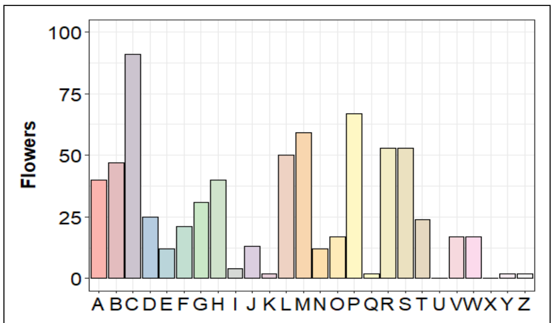


Figure 1: Language of Flowers: distributed flower types arranged in alphabetically sorted buckets of names. Buckets U and X are evidently empty.

To visualize the distribution of the vocabulary from several different viewpoints, we used R (R Core Team, 2013) to render a word cloud (Figure 2) that depicts the top 150 frequent sentiment labels in the dictionary of the language of flowers. Noticeably of the highest occurrence count are words of emotional romantic connotations like 'love', 'beauty', 'affection', 'friendship', and 'heart'. In Figure 3, we follow to provide term frequencies for each of the top 25 sentiment labels in the vocabulary, as the words 'love', 'beauty', and 'affection' occur 40, 31, and 12 times, respectively. Lastly, the distribution of sentiment word lengths is of notable importance to assess flower prediction performance. To this extent, Figure 4 highlights 330 single-word, 100 four-word, and 80 two-word long sentiment sentences, with a maximum sequence length of eighteen words.

Table 1: Language of Flowers: a sample of sentiment phrases each identified with a single ground-truth flower-type. Flower names of multiple words are hyphenated to form a compound.

Sentiment Phrase	Pivotal Flower
endearment, sweet and lovely	carnation-white
you pierce my heart	gladiolus
pleasant thoughts, think of me	pansy
joy, maternal tenderness	sorrel-wood
enchantment, sensibility, pray for me	verbena

Figure 2: Language of Flowers: word cloud of the top 150 frequent sentiment labels. Font size is proportional to the number of word occurrences in the corpus, with 'love', 'beauty', and 'affection' leading.

Despite the relatively concise vocabulary of size $|V| = 1{,}386$ tokens, rather than to use a 1-of-$|V|$ sparse representation of a one-hot vector $\in \mathbb{R}^{|V| \times 1}$, we map one-hot vectors onto a lower-dimensional vector space using the Word2Vec (Mikolov et al., 2013b) embedding technique that encodes semantic word relationships in simple vector algebra. To train word vectors effectively, we enabled negative sampling in both the skip-gram and continuous-bag-of-

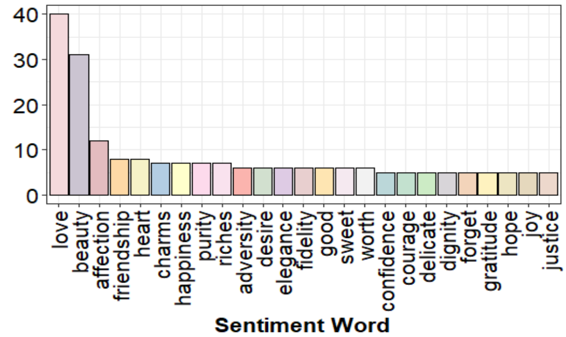


Figure 3: Language of Flowers: distribution of the top 25 frequent sentiment labels in the vocabulary.

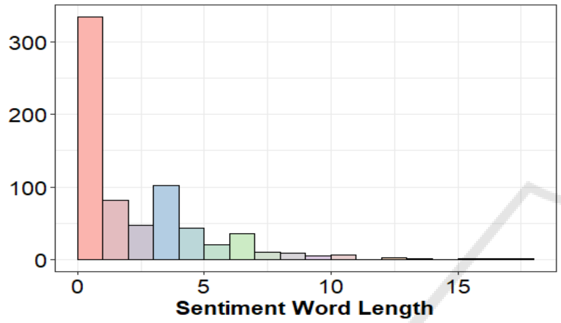


Figure 4: Language of Flowers: distribution of sentiment word lengths across the entire train set.

words (CBOW) neural models, and found word vector dimension, d, a critical hyperparameter to tune and yield consistent flower-selection predictions in RNN.

3 FLOWER CANDIDACY

Formally, we represent the language of flowers as a list of translation records, each defines a pair of a source sentiment phrase and a unique target flower. We use the notation $v_{1:k}$ to describe the sequence $(v_1, v_2, \ldots, v_k)$ of k vectors and correspondingly denote a translation pair as $(s_{1:k}, f)$. A sentiment-flower pair is decomposed into a k-length sentiment input $s_{1:k}$ we linearly stream into RNN and an output concatenation $(s_{2:k}, f)$ of k word vectors, and establish a ground-truth pivotal relation between a flower type and its immediate preceding context. We note that our language corpus incorporates short-text sentiment phrases of length k that ranges from one to eighteen word vectors (Figure 4). To further align our rendition interface with the RNN architectural notation, we let $x_t \in \mathbb{R}^d$ be the d-dimensional word vector identified with the t-th word in a text sequence, and denote an end-to-end translation record as $(x_1, x_2, \ldots, x_k, x_{k+1})$, or more compactly $x_{1:k+1}$. We use the Gated Recurrent Unit (GRU) (Cho et al., 2014; Chung et al., 2014) variant of RNN that adaptively captures long term dependencies of different time scales. At each time step t, the GRU takes an input word vector x_t and the previous n-dimensional hidden state h_{t-1} to produce the next hidden state h_t. Conceptually, the forward GRU has four basic functional stages that are governed by the following set of formulas:

$$
\begin{aligned}
z_t &= \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}) \\
r_t &= \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}) \\
\tilde{h}_t &= \tanh(r_t \odot U h_{t-1} + W x_t) \\
h_t &= (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}
\end{aligned}
$$

where $W^{(z)}, W^{(r)}, W \in \mathbb{R}^{n \times d}$ and $U^{(z)}, U^{(r)}, U \in \mathbb{R}^{n \times n}$ are weight matrices, and the dimensions n and d are input configurable hyperparameters. The symbols $\sigma()$ and $\tanh()$ refer to the non-linear sigmoid and hyperbolic-tangent functions, and $\odot$ is an element-wise multiplication. A backward fed GRU defines $\overleftarrow{h}_t = \overleftarrow{GRU}(x_t), t \in [k, 1]$.

In modeling the language of flowers, our main objective is to score flower candidacy for a bouquet, by predicting flower type probabilities based on the short-text sentiment sequence retained in each translation record. Serialized sentiment vectors $x_{1:k}$ are streamed to RNN in isolation and independently, but the inherent persistent-memory nature of GRU summarizes at each time step the newly observed word vector in a record sequence, x_t, with the cumulative previous context, h_{t-1}. Similarities computed as inner-products between each of the input converted word-vectors and the GRU encoded h_t are then forwarded to a softmax discriminative decoder. Hence, the next predicted word is the output probability distribution $\hat{y}_t = \text{softmax}(W^{(S)} h_t)$ over the entire language vocabulary, where $W^{(S)} \in \mathbb{R}^{|V| \times n}$ and $\hat{y}_t \in \mathbb{R}^{|V|}$, and $|V|$ is the cardinality of the vocabulary. During the training of RNN, we attach a higher scoring bias to the ground-truth pivotal flower-word, x_{k+1}, that immediately succeeds the last word of a sentiment phrase, x_k. Whereas in evaluation, to generate a selection for a bouquet of flowers we rank all the posterior probabilities of $\hat{y}_t$ that were predicted for x_k, and from the top-m indices we filter out all sentiment word instances, and return to the user the remaining flower tokens.

4 SHARED REPRESENTATION

In their recent work, Lee and Dernoncourt (2016) show that the chronology of sequences of short-text

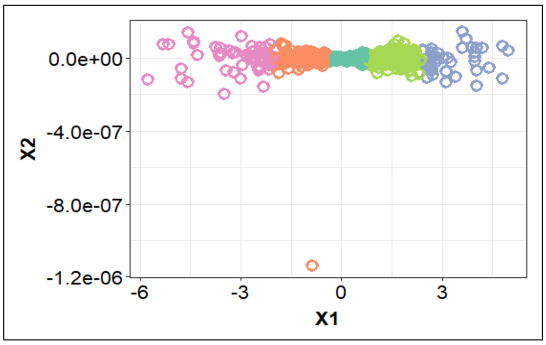


Figure 5: Language of Flowers: combining Multidimensional Scaling and k-means clustering to visualize sentiment phrase relatedness in semantic vector space. Shown five collections with only a few outliers.

representations fed to RNN, improves sentence classification accuracy. Motivated by their results, we sought after a sequencing model that schedules sentiment sentences to RNN based on plausible semantic relatedness between phrases. However, our online acquisition of translation records that are enumerated in alphabetical bins based on flower names (Figure 1), implied a partition type that constrains the more interesting information about sentiment semantic similarity.

To address this shortcoming, we first had to avoid non-conforming representations in dot-product similarity computations by reshaping the varying dimensionality of sentiment clauses into uniform sized feature vectors. We chose to leverage a basic convolutional architecture practice and applied mean pooling to each of the encoded sentiment sequences $x_{1:k}$, and averaged the word vectors to yield a single vector representation $s = \frac{1}{k}\sum_{t=1}^{k} x_t$, where $s \in \mathbb{R}^d$. From the set of single-vector formatted sentiments $s_{1:l}$, where $l = 701$, we follow by constructing a distance matrix $D \in \mathbb{R}^{l \times l}$, and use Multidimensional Scaling (MDS) (Torgerson, 1958; Hofmann and Buhmann, 1995) to project the large dissimilarity matrix onto a two-dimensional embedding space. By combining MDS with k-means (Kaufman and Rousseeuw, 1990), we produced visualization of clusters that group semantically close sentiment phrases (Figure 5). Surprisingly, only a few outliers persist and most sentiment sequences notably gather consistently. Results we report next were obtained by scheduling sentiment sequences to RNN in the order prescribed by their projected coordinates, from left-to-right and bottom-to-top. This is motivated by sharing representation power of similar context, where the final hidden state of the current encoded sentiment is fed as the initial hidden state of the next spatially closest sentiment sequence.

5 EVALUATION

Our workflow for evaluation is straightforward. First, the user enters our system a desired count of plant types ranging from 1 to m, along with an arbitrary sentiment composition of words that are part of the vocabulary of the language of flowers (Figure 2). The user text sequence is transformed to a single word-vector representation and follows cosine similarity calculations (Salton et al., 1975; Baeza-Yates and Ribeiro-Neto, 1999) with each of the trained mean-pooled sentiment-sentences, s_i. For the semantically closest translation pair, we query the pivotal and supporting flowers and return to the user distinct flower images and proportional weights that are used for final bouquet arrangement. To compare the predictive performance of our GRU-based RNN system against, we chose a baseline that we feed with our mean-pooled representation of a sentiment sentence, and use softmax regression for the unsupervised learning algorithm. We cross validated our baseline on both a held out development set and an exclusively generated test-set. Figure 6 illustrates an architectural overview of the pipelines for both the main and baseline computational paths.

5.1 Experimental Setup

To evaluate our system in practice, we have implemented our own versions of a GRU-based RNN module and the Word2Vec embedding technique, both natively in R (R Core Team, 2013) for better integration with our software framework. Considering our self-sustained corpus of irregular context, we chose to collectively initialize our word vectors randomly and learn them purely from the dictionary data. Instead of obtaining pre-trained word vectors on large vocabularies of external corpora that miss most our flower names. Skip-gram and CBOW performed almost identically in terms of flower prediction accuracy, both are challenged by a language that combines short-text sentences with a small number of samples, respectively. Our model is trained to maximize the log-likelihood of predicting the ground-truth target flower for the language set of translation records, using mini-batch stochastic gradient descent (SGD) and performing error back-propagation. At every SGD iteration, GRU weight matrices and word vectors are updated, using the AdaDelta parameter update rule (Zeiler, 2012).

In Table 2, we list our experimental choices of hyperparameters that control both the RNN and Word2Vec subsystems. Given the succinct nature of text fragments that constitute our sentiment phrase,

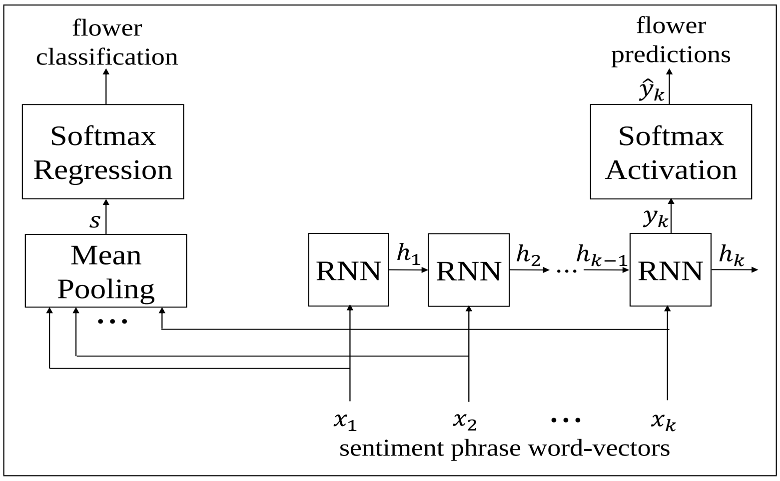


Figure 6: Architecture overview of computational pipelines for the mainline feedforward GRU-based RNN (right) and softmax regression baseline (left).

Table 2: Experimental choices of tuned hyperparameters applied to both RNN and Word2Vec modules.

Hyperparameter	Notation	Value
Hidden state dimension	n	18
Word vector dimension	d	100
Context window	c_w	5
Negative samples	n_s	10
Mini-batch size	b	10
Top ranked probabilities	m	10

we held the context window size c_w at five words symmetrically, and assigned the number of negative samples n_s to its recommended default value for small datasets (Mikolov et al., 2013b). Correspondingly, we allocated a reasonable example fraction of our train set size for the SGD mini-batch size b. To optimize the score for predicting the flower types, we modified one dimensional parameter at a time and kept the other fixed. This culminated in setting the preferred hidden-state dimension n to the maximal word length of a sentiment phrase (Figure 4), as we noticed accuracy performance improvement by 6.7 percentage points when modifying the word vector size d from 10 to 100. However, we observed a diminishing return in embeddings larger than the hundred dimensions. Unless stated otherwise, the results we report were obtained using the skip-gram neural model and a unidirectional GRU-based RNN, with $m = 10$.

5.2 Experimental Results

We chose a simple baseline model that replaces RNN with softmax regression for learning, and as feature vectors uses a sentiment representation of mean-pooled word embeddings, $s \in \mathbb{R}^d$. Each short-text sentiment sequence of the train set is assigned one-of-five categorical labels (Figure 5), as we turn our problem to perform multi-class flower classification.

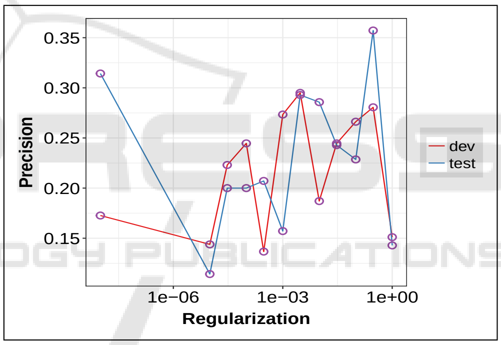
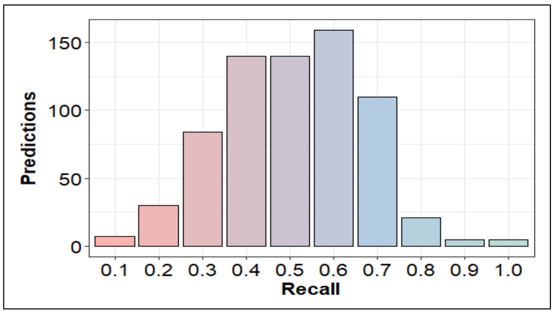
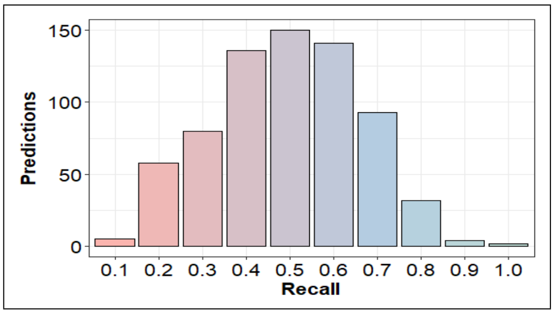


Figure 7: Baseline softmax regression results on an inclusive development set and exclusive test set. Showing precision as a function of a logarithmic-scaled regularization parameter.

For validation evaluation, we formed a two-way data split of 80/20 by proportionally sampling random sentiment sentences from each of the five training classes into a held out development set. To construct our test set and allow for the study of our problem domain itself, we leveraged the generative capacity of an RNN-based language model (Sutskever et al., 2011), and created entirely new text sequences that closely resemble the emotional context at the core of the language of flowers. The test collection we built comprises 140 synthetic sentiment sentences of variable length, about 20 percent of the train set size, and is evaluated against the unsplit train-set size of $l = 701$. Each of these extended sentiment phrases is paired with one of the existing flowers in the corpus and hence implicitly inherits a ground-truth multi-class label. The pairing of a flower to a generated test



(a) Forward RNN



(b) Backward RNN

Figure 8: Distribution of flower predictions that are sampled across the entire train set at discrete recall steps. Contrasting (a) forward $x_{1:k}$ with (b) backward $x_{k:1}$ propagation of a sentiment phrase in RNN.

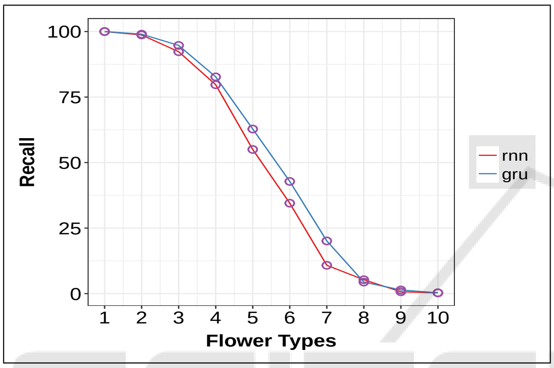


Figure 9: Comparing standard to GRU-based RNN on percentage recall of flower predictions as a function of increased flower type count $\in (1,2,\ldots,m=10)$. Curves present a non-linear decline in performance.

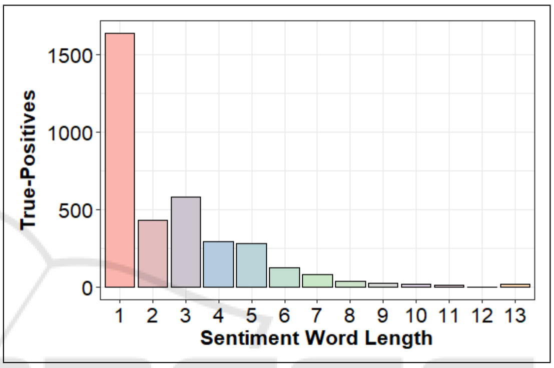


Figure 10: Accumulated true-positives of flower predictions as a function of the word length of the sentiment phrase. The data produced matches the vocabulary distribution shown in Figure 4.

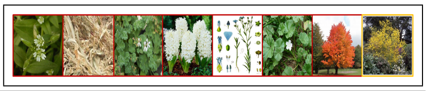
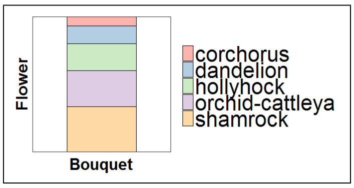
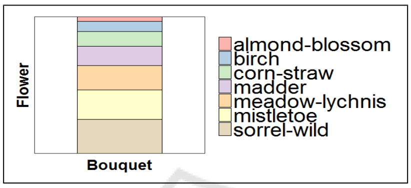
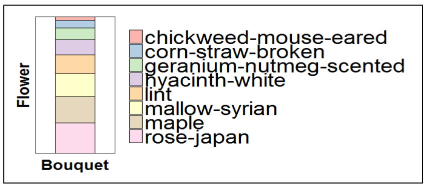
phrase g is obtained by computing the similarity with each of the train set sequences s_i, and finding the closest record in $\text{argmax}_{1 \leq i \leq l} \text{sim}(g, s_i)$. Figure 7 shows our baseline performance results as a function of a logarithmic-scaled regularization parameter, on both the development and test data sets with an average precision of 0.22 and 0.23, respectively.

In evaluating our GRU-based RNN system, our main goal is to quantitatively assess the quality of flower predictions looked at from differing perspectives. The top-m ranked probabilities presented in the network output layer may index any combination of flower and sentiment token types. We let t_f denote the number of flower tokens and t_s the sentiment token count in this m-sized window of descending probabilities. Thus, the performance interpretation of t_f and t_s in our model amounts to the true-positive and false-negative observations made in each flower selection query, respectively. For our flower prediction metric, we follow the recall measure of relevance that is defined as the ratio $\frac{t_f}{t_f+t_s}$, or more compactly $\frac{t_f}{m}$.

Figure 8 provides distribution of flower predictions in recall bins of 0.1 increments for both forward and backward sentiment-phrase propagation in RNN (Schuster and Paliwal, 1997), and in Figure 9, we translate the discrete data to a continuous recall curve as a function of a non-descending number of flower types in a bouquet. The average recall for the sweet-spot content of one to five flower types is 87.8% for GRU-based, and 81.5% for plain RNN, both outperforming the softmax regression baseline by a factor of 3.8X and 3.5X, respectively. In Figure 10, we present the collective number of true-positives as a function of a non-descending word length of the sentiment text sequence. Flower predictions closely correlate with the word length distribution in the vocabulary (Figure 4), mainly owing to streaming sentiment context to RNN in an order prescribed by spatial proximity in semantic vector space.

High-end bouquets created of tens of flower types are considered a rarity, but despite their limited presence and small floral market-share, they make an important case for our system evaluation. In Figure 11, we review the distribution of flower predictions for high-end bouquets with the system hyperparameter $m = 50$. Distinctly, incidents of unpredictable bouquet selections do transpire and display a fairly sym-

Table 3: Visualization of statistically created flower selections that were prompted by user-supplied sentiment expressions, shown at the top. We outline side-by-side three bouquets of five, seven, and eight flower candidates, respectively. Depicted are thumbnail images for both the pivotal flower, on the far right, and top-ranked supporting flowers to its left, along with individual flower names and their proportional mix weights.

	Sentiments	
light heartedness	*wit ill-timed*	*beauty is your only attraction*
	Flowers	
		
	Proportions	
		

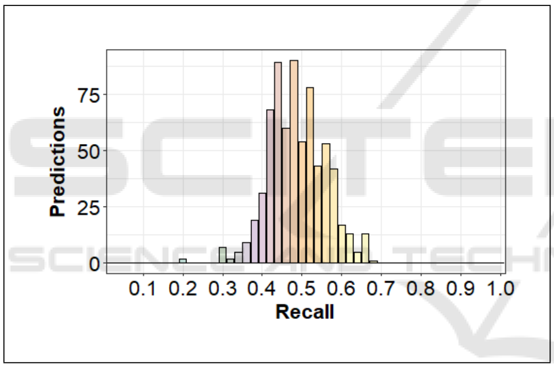


Figure 11: Distribution of flower predictions for high-end bouquets ($m = 50$). Shown sampled at discrete recall steps of five flower type bundles, as unpredictable selections transpire symmetrically.

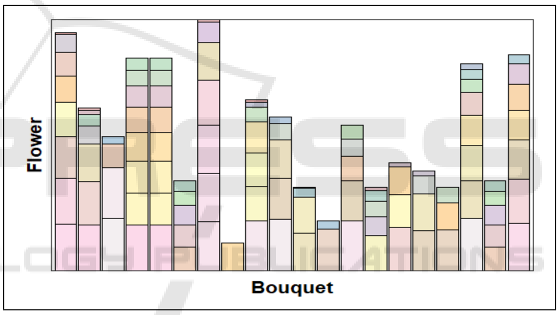


Figure 12: A random sample distribution of sentiment sensible bouquets, each shown with proportional weights of an assembled size that ranges from one to eight flower types.

metrical behavior with sizes ranging from 1 to 14, excluding a bouquet size of ten plants with two predictions, and from 35 to 50 assuming no predictions. Conversely, for bouquets of mid-range sizes from 20 to 29 species that possess each at least twenty predictions, the total predictions amounts to 608 and represent about 86.7 percent of the training set dimension. We note that unpredictable selection sizes are of very low probability for mainline bouquets ($m \leq 10$), and for high-end bouquets, to ensure the requested number of flower types issued by the user is satisfied, our system supplies the user a precompiled list of preferred bouquet sizes to chose from, for each m.

In Table 3, we present end-to-end visualization of our approach to statistically-generated flower selections from decoded sentiment clauses. Outlined side-by-side are three bouquets of five, seven, and eight flower candidates, respectively. The incoming short-text sentiment sequences are first evaluated for the most similar sentiment phrase retained in one of the language translation records to enter our pipeline, shown at the top. We then depict compositional thumbnail images of the pivotal flower, placed on the far right, and the supplemental flowers to its left, along with individual names of plants and their respective proportional blend-weights. Stacked relatively, slices are shown in a non-ascending manner, starting from the principal flower at the bottom and up to the lowest ranked flower. In Figure 12, we highlight a sample distribution of twenty sentiment-perceived bouquets, each with a unique center-of-attention flower. Selections are shown for varying sizes ranging from one to eight plant types.

6 CONCLUSIONS

In this work, we have demonstrated the plausible potential in composing a bouquet made of a lead and supporting flowers by attending to a set of unqualified sentiment expressions we stream over RNN. We confirmed that GRU-based RNN improved our flower prediction quality by about ten percentage points compared to a standard RNN, and feeding RNN with spatiotemporal sentiment context prove particularly beneficial to the performance of short-text sequences. Yet using bidirectional propagation of sentiment word vectors that enter RNN was less instinctive, and contributed to an inconsequential prediction gain. Our proposed simple workflow offers on average high prediction recall to hundreds of mix choices for a mainline bouquet, and sends a more cohesive emotional message that is made of semantically related sentiments.

To the extent of our knowledge, the work we presented is first to apply computational linguistic modeling to the language of flowers. We contend that floriography is an important NLP discipline to pursue from both its rooted historical impact on society culture, and the prospect to influence areas of critical theory and sentiment analysis. In its current state, the corpus we used in this paper is small and challenging, but we anticipate the language to expand sentiment translation to thousands of flower plants and further merit our statistically reasoned system. A direct progression of our work is to evolve to a task that matches flower-sentiment pairs from unstructured full text and not just from a set of prescribed sentiment phrases, and have a profound practical importance to impact a much broader scope of application domains that include cryptography and secured communication.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their insightful suggestions and feedback.

REFERENCES

Baeza-Yates, R. and Ribeiro-Neto, B., editors (1999). *Modern Information Retrieval*. ACM Press Series/Addison Wesley, Essex, UK.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259. http://arxiv.org/abs/1409.1259.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555. http://arxiv.org/abs/1412.3555.

Diffenbaugh, V. (2011). Victoria's dictionary of flowers. http://aboutflowers.com/images/stories/Florist/languageofflowers-flowerdictionary.pdf.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Hoang, C. D. V., Cohn, T., and Haffari, G. (2016). Incorporating side information into recurrent neural network language models. In *Human Language Technologies: North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1250–1255, San Diego, California.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hofmann, T. and Buhmann, J. (1995). Multidimensional scaling and data clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 459–466. MIT Press, Cambridge, MA.

Kaufman, L. and Rousseeuw, P. J., editors (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, NY.

Lee, J. Y. and Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. In *Human Language Technologies: North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 515–520, San Diego, California.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781. http://arxiv.org/abs/1301.3781.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *International Speech Communication Association (INTERSPEECH)*, pages 1045–1048, Chiba, Japan.

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 (NIPS)*, pages 3111–3119. Curran Associates, Inc., Red Hook, NY.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/.

Roof, L. and Roof, A. H. (2010). Language of flowers project. http://languageofflowers.com/index.htm.

Salton, G. M., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Signal Processing*, 45(11):2673–2681.

Sutskever, I., Martens, J., and Hinton, G. (2011). Generating text with recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pages 1017–1024.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112. Curran Associates, Inc., Red Hook, NY.

Torgerson, W. S. (1958). *Theory and Methods of Scaling*. John Wiley and Sons, New York, NY.

Tran, K. M., Bisazza, A., and Monz, C. (2016). Recurrent memory networks for language modeling. In *Human Language Technologies: North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 321–331, San Diego, California.

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701. http://arxiv.org/abs/1212.5701.