

# A Review of the Unified Corpus and a Methodology for Improvement on Generalizable Emotion Detection

Jacob Fuehne

Department of Linguistics

University of Illinois at Urbana-Champaign, USA

jfuehne2@illinois.edu

## Abstract

Emotion detection and classification in natural language processing is an important part of improving overall understanding of language and quality of interaction with computer systems. It is impossible to completely control the variations of text that a real world classifier may face, and so a generalizable approach is necessary. In this paper, I focus on a literature review of the Unified Corpus paper and I outline relevant techniques for the purpose of generalizable emotion detection. The contributions of this paper are a review of results of the Unified Corpus, improvements in ability to further iterate on the Unified Corpus code, and a review of SOTA techniques in ensemble methods for generalizable emotion detection.

## 1 Introduction

Emotion detection is an important topic in computational linguistics as it relates to computers ability to interpret and interact with people. Emotion detection systems are often based on psychological models as they focus on emotion categories. Emotion detection has applications in dialog systems, business analytics, mental healthcare diagnosing, and data mining. Research in this field has focused mostly on polar sentiment analysis in the past, with only recently work being done on classifying a broader range of emotions. A less common point of research in affective computing is done on generalizable approaches to emotion detection.

Previous work in applying emotion detection techniques to generalizable test data include Bostan and Klinger's 2018 paper, where they develop and publish a new dataset, the Unified Corpus, composed of over a dozen publicly available corpora (Bostan and Klinger, 2018). While data is abundant in modern times, emotion annotated data is sparsely available for public use, and manually annotating datasets requires resources that researchers

may not have access to. Thus, it is important for researchers to publish datasets and find ways to leverage existing datasets for advancement in the field. For domain specific tasks, or tasks relating to particular linguistic registers of a language, the best option has been shown to be a specialized corpus. However, the boundaries for how specialized a corpus needs to be to show improvements are not clearly defined, and current statistical metrics for comparing corpora are inadequate for capturing more abstract variations such as linguistic registers and linguistic variations. These methods fall short of measuring the usage of language and are often simply a rough approximation. Combining topic modelling approaches such as LDA with emotion detection has shown promise, as was done in the (CITATION), however, LDA can only capture topic domains by statistical means, and even in situations where the domains are captured perfectly, topic and domain does not directly correspond to linguistic variation and how language is used.

In the results of Bostan and Klinger's paper, they found that classifiers trained on the Unified Corpus consistently underperformed classifiers trained on a much smaller part of the corpus. In this paper, I perform an analysis of Bostan and Klinger's paper and results, as well as a literature review of relevant work to their task, and I outline a methodology for improvement based on ideas of previous papers. I propose a methodology of future work in applications that support generalizable emotion detection through the use of multiple classifiers trained on corpora of varying similarity. Due to certain aspects that I will mention, there were limitations surrounding my ability to fully test the hypothesis proposed. As such, I offer background research of proof of concept works in this area for aspects that I have was unable to provide my own experimental data for, as part of my literature to support the methodology. Also provided in this paper are

pre-trained classifiers for each of the trials tested in this paper. Project outputs are included as jupyter notebook files and are viewable through the commit history of the project, as well as in a collection of github results files.

## 2 Corpus Design Practices

Examining the various aspects of a proposed generalizable, ensemble approach to emotion detection, one must also take an ensemble approach to understanding the process and look at different parts.

### 2.1 Corpus Design Practices

At the abstract level, corpora are typically selected based on a broad categorical definition that they were collected under, such as a twitter corpus gathered through the Twitter API like the Twitter Emotion Corpus by (CITATION), or by a topic that unites the entries, such as the Tales corpus by (CITATION). Datasets are often created to complement the work of some other task in a publication, and the shared characteristic among them is predominantly defined with this goal in mind. Some exceptions to this include datasets that are intended to be representative as part of a challenge, such as the SemEval datasets like SSEC. In the case of the authors of the Unified Dataset, the goal was to gather many publicly available datasets and create a methodology and application for the purposes of uniting the various corpus formats. Thus, the dataset has both large amounts of variation, while at the same time having disproportionate representation, as some included data sources are much larger than others, and each source has its own internal differences.

### 2.2 Domain, Topics, Registers, and Linguistic Variation

With the exception of corpora such as the REMAN corpus by Kim and Klinger, researchers often don't publish subcategorical information such as book genres, or in the case of twitter, possibly a topic model. This makes it difficult to compare corpora directly, as notable vocabulary differences can occur with subgenres, and the linguistic variations of speakers within a genre can produce opposing meanings for words. A word that can have two opposing or contradictory interpretations is known as a contronym. While many classification approaches such as simple bag-of-words can achieve what would be considered acceptable accuracy in

some contexts through simply using probability and the most commonly occurring definition, one would expect such a probabilistic approach to be consistently wrong in contexts with opposing meaning. An example of this is illustrated in the following Arabic example that is outlined in Saadany and Orasan's paper (CITATION) where they explain a contronym between Dialectical Arabic (DA) and Modern Standard Arabic (MSA):

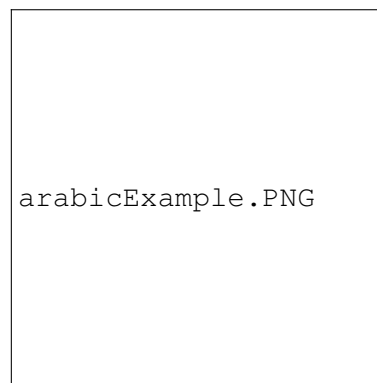


Figure 1: "The narration is terrible, it's only flaw is the last part." or "The narration is great, it's only flaw is the last part."

This sentence is translated as, "The narration is terrible, it's only flaw is the last part." However, this is because the word for terrible in this sentence means terrible in MSA, but it often means great in DA. Thus, the sentence would actually be "The narration is great, it's only flaw is the last part." if the speaker is using DA. Variations such as these cannot be adequately captured from simple statistical techniques.

### 2.3 Corpora comparison metrics

In order to compare corpora, statistical approaches are used in spite of their inability to capture certain linguistic variation features in the absence of any better metric. Critical to the development of an ensemble method that generalizes emotion classification is including multiple corpora and grouping them based on similarity. As previously mentioned, there is an unknown threshold for how similar corpora must be for their concatenation to result in a performance increase. Due to varying amounts of linguistic representativeness within each corpus and the lack of an ability to definitively compare corpora, even by humans, it is difficult to even rate the effectiveness of corpus comparison metrics by any other means than combining them and evaluating performance. For this task of defining

which corpus comparison metrics are best, a gold standard benchmark was proposed by Kilgariff in 2001, called Known-Similarity Corpora.

$$\sum_{i=1}^n (n-i) \left( \frac{i(i+1)}{2} - 1 \right)$$

For this metric, two distinct text types are compared and split into portions A and B. Then a set of corpora are created such that corpus 1 is composed of 100% A, corpus 2 is composed of 90% A and 10% B, corpus 3 is composed of 80% A and 20% B, and so on. By having known proportions of distinct text types in each, one can evaluate what a corpus comparison metric would predict that proportion is. As mentioned with internal linguistic variation issues, it is important that the distinct text types that are chosen will properly isolate linguistic variation, or at the least, text types that are distinct for the purposes of one's task. Just like feature sets, corpus comparison metrics are not created equal, and there is not a universally best SOTA approach. For example, in the same 2001 paper where Kilgariff proposed the Known Similarity Corpus method, he found that the Chi-Square test was most suitable at the time for comparing corpora. However he admitted that sampling size was a serious concern with Chi-Square test, represented below:

$$\sum \frac{(O - E)^2}{E}$$

In the Chi Square test for corpora comparison, token frequency values across documents/text samples are gathered as a bag-of-words and represented as an MxN matrix, where it is text sample by term frequency. To get the expected value E, for each term, one will sum the observed values for token frequencies in the column and multiply this sum by the sum of the that term's frequencies in other text samples. The product of this is then divided by the total number of words. Once this value is acquired, the E is applied to the equation above. This gives the Chi-Square value for each term frequency. To get the final Chi-Square value, one must then sum all calculated chi-square values. Then resulting value is compared to a chart of critical values for the number of degrees of freedom used in the Chi-Square test. This method has the added bonus of having more meaning behind the comparison, as it can show statistical significance and prove the null hypothesis.

The limitation that Kilgariff acknowledged in their paper is that in order to be applicable when comparing the Brown Corpus with the LOB Corpus, he had to agglomerate texts into 10 samples comprising 50 texts and 100,000 words each. Special care had to be taken so that samples had very few zero value occurrences. However, in the case of emotion detection, many corpora are manually annotated, with the exception of distantly supervised collection. And as such, while Chi-Square is a suitable metric for big data, it may not be applicable to many cases in affective computing.

The limitation that Kilgariff acknowledged in their paper is that in order to be applicable when comparing the Brown Corpus with the LOB Corpus, he had to agglomerate texts into 10 samples comprising 50 texts and 100,000 words each. Special care had to be taken so that samples had very few zero value occurrences. However, in the case of emotion detection, many corpora are manually annotated, with the exception of distantly supervised collection. And as such, while Chi-Square is a suitable metric for big data, it may not be applicable to many cases in affective computing.

Cosine Similarity is a metric that is very commonly used in corpus comparison perhaps primarily for its simplicity. The metric is simply the dot product of a normalized vector of term frequencies between corpora. The authors of the Unified Corpus analyzed the dataset for cosine similarity, however, they did not publish their particular implementation with their code, nor did they include descriptive details on the settings used for cosine similarity.

In the 2020 paper from Lu, Henschion, and Namee, variants of Jensen-Shannon distance have been shown to be the most effective for distinguishing corpora in the Known Similarity Corpus method. Through analysis, it was found that JSD-pechenick, shown below, is the recommended variant methodology for its performance in diverging datasets and for its resilience to corpus size, making it suitable for a domain such as emotion detection. While not the variant proposed by pechenick, there are publicly available libraries with JSD, such as scipy. However, late in the work of this paper, it was found that the method of storing corpora was incompatible with JSD, and the implementation of JSD was left commented out in the code left as future work.

Lastly, and closest to capturing a generalizable

approach to emotion classification is that of the models that combine topic modeling through Latent Dirichlet Allocation with emotion classifiers in an ensemble approach. A recently published paper on this proposed a model framework where an LDA classifier was used to extract topic features, and then use the extracted topic features as training input into a classifier of the user's choice, in the paper choosing to use the sklearn SVM model as a baseline.

## 2.4 The Unified Corpus Paper

While the original paper, An Analysis of Annotated Corpora for Emotion Classification in Text, found that when using their unified corpus as training data it led to worse performance than considerably smaller but more closely related datasets, this paper served as a foundation for future work to generalizable emotion classification and serves as the initial inspiration for this paper. As you can see from Table X, the dataset consists of 14 corpora, but only 13 of them were included in the publicly available version (Blogs is available by request only). And of those 13, only 11 are annotated for emotion, with fb-valence-arousal being only annotated for valence and arousal, and EmoBank only annotated for VAD. Descriptions of the datasets can already be found summarized in the original paper, or in full through the cited links.

In the original paper, the authors merged emotion labels to a mapping that roughly follows Plutchik and Parrott (CITATION see table 4). The Unified Corpus contains the emotions anger, anticipation, confusion, disgust, fear, joy, love, noemo, sadness, surprise, and trust, with the most common annotation metric being Ekman's model of basic emotions, which includes anger, disgust, fear, joy, sadness, surprise. As one can see by viewing the table, and seeing the domains and topics of the included corpora, the unified corpus is composed of many different linguistic registers, and with varying ranges in size. The largest of the datasets included, Crowd-Flower and TEC are both tweets of general topic. Much of the twitter based corpora included in the unified corpus (the most common domain) contain a large linguistic register range, with thankful tweets about family to sports to complaints and many vulgar tweets from different linguistic variations.

The unified corpus is not available for direct

download, and it must be compiled into a .jsonl file using the code provided in the original paper's github repo. For the purposes of the paper, the code is included in the linked github out of convenience, but if this were an ACL submission, it would likely be impossible to do so due to licensing restrictions surrounding distribution and republication.

## 3 Analyzing the Unified Corpus Paper

To analyze the Unified Corpus paper's results, I ran the project as published on github initially. However, I found immediately that the project had issues with implementation. Notably, the initial code to load in files had typos and did not function as intended, and so I rewrote the means for conducting trials, completely ditching the .py format of the original in favor of a jupyter notebook approach that would allow for easier visualization of outputs.

The authors of the original paper claim to have used a MaxEnt model with 10 fold cross validation and bag-of-words features, in this case based on their code it is safe to say that by MaxEnt they are referring to the sklearn logistic regression classifier. This classifier is claimed to have been used in trials where both corpora are singly labelled (ie, 1 emotion is marked true and all others are marked 0) and in trials where one of the corpus sources is multi labelled, a OneVsRestClassifier from sklearn is used, constructed from the same logistic regression classifier. In order to try and replicate the results as closely as possible, I initially conducted my trials with their code, but I later found more issues as I tried to progress. Another issues encountered was that the Electoral-Tweets dataset is not compatible with their current implementation of their project, as Electoral-Tweets will have some situations where it is marked to have an emotion, but it is also marked with confusion, despite that this data is marked explicitly as single emotion labelled. Thus, in trials conducted with electoral-tweets, an exception will occur anytime this dataset is run without forcing multilabel classification. An example of such a data entry is shown below:

---

```
{ "id": 175096,
  "VAD": {
    "valence": null,
    "arousal": null,
    "dominance": null
  },
  "source": "electoraltweets",
  "text": "I immediately doubt the sanity
of someone who is okay with taking
responsibility for all of America's
```

Dataset	Domain	Topic	Usable	Source
AffectiveText	headlines	news	✓	Strapparava (2007)
Blogs	sentences	blogs	✗	Aman (2007)
CrowdFlower	tweets	general	✓	Crowdfower (2016)
DailyDialogs	dialogues	multiple	✓	Li et al. (2017)
Electoral-Tweets	tweets	elections	✗	Mohammad (2015)
EmoBank	sentences	multiple	✗	Buechel (2017a)
EmoInt	tweets	general	✓	Mohammad (2017b)
Emotion-Stimulus	sentences	general	✓	Ghazi et al. (2015)
fb-valence-arousal	facebook posts	questionnaire	✗	Preotjuc (2016)
Grounded-Emotions	tweets	weather/events	✓	Liu et al. (2017)
ISEAR	descriptions	events	✓	Scherer (1994)
Tales	sentences	fairytale	✓	Alm et al. (2005)
SSEC	tweets	general	✓	Schuff et al. (2017)
TEC	tweets	general	✓	Mohammad (2012)

Table 1: List of corpora used in the Unified corpus and their domains and topics, as well as a marker for whether or not the data is actually usable in emotion classification

```

hopelessness... #president
#dirtyjob",
"emotions": {
    "joy": 0,
    "anger": 0,
    "sadness": 0,
    "disgust": 0,
    "fear": 0,
    "trust": 0,
    "surprise": 1,
    "love": null,
    "noemo": 0,
    "confusion": 1,
    "anticipation": 0,
    "shame": null,
    "guilt": null
},
"split": null,
"emotion_model": "Ekman+ET",
"domain": "tweets",
"labeled": "single"
}

```

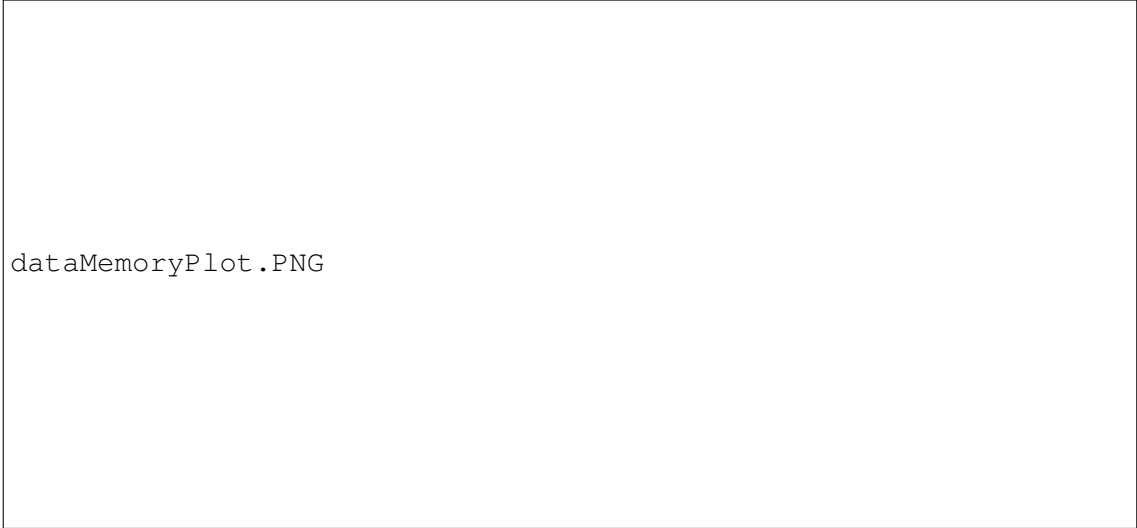
As such, in my own personal run of the program, I decided to throw out the electoral tweets dataset for individual trials.

It is also worth mentioning that a core component of the Unified Corpus paper’s authors original code is that in their implementation they create a top 5000 bag of words for each entry in the source corpus, and they do the same for the test data. Thus, the size of their numpy arrays is actually much larger than that of the raw text, despite it being reduced to 1’s and 0’s. Due to numpy’s clever optimization, the effective size of a numpy array is not so simple to calculate, and there is not a simple formula for getting the size other than knowing that it is at worse  $O(mn)$ . Through the use of

`sys.getsizeof()` and `numpy.nbytes`, I was able to obtain sizes for raw text and numpy arrays throughout my application runs. I found that with this implementation, the size of the algorithm actually scales by more than 2000x the size of the raw text. Thus, to run something such as the All-Vs trials, you would need to work with the numpy array variable that is over 4 gigabytes. A graph of the recorded data memories is seen below, and archived outputs of these data metrics can be found in the commit history of the github project. 2 [1-2] 2 [3-4]

It is worthwhile to note that the data memory sizes shown in table X reflect only those of either the train set or the test set numpy arrays, and not the variable size for the application. With the program needing to load in the train set, test set, raw text, the classifier, and then generate the prediction label set, data memory adds up quickly. While my computer has a larger than average amount of RAM, I found myself on occasion reaching 99% RAM usage for my application, and before memory optimizations, I had issues with jupyter notebook crashing.

In doing these trials, I took special care to document and automate my trials. I used a library called `joblib` to produce pickle files of my classifier variables and I used `numpy.save` to save the numpy arrays. This allows me to simply load a file containing the saved variable as it existed in python whenever I need to, thus avoiding the many hour wait to test my application after non-fundamental changes to the code. The results of each trial can be found in the github repo for the project. Attached



dataMemoryPlot.PNG

Figure 2: This graph plots the data memory size of the raw text against the size of the numpy array

in a google drive are the classifiers for each trials. It was also my intention to publish the numpy arrays for my project for the purpose of full replicability, but unfortunately, after completing all the trials, I found that the total size of all files was 151 gigabytes of pickle and numpy files, which exceeds the capabilities of my data hosting. If the application is run locally, however, pickle values can still be stored without issue.

Through my I ran the application and stored the results of each, allowing the application to run at night or while I was working on other aspects of the code. While I did not explicitly time the application, it often took several hours to run larger trials, and it wasn't until near the end of this project that I was able to complete a successful run of the data intensive trials. This is also in part because the original code for conducting all-vs and the code for conducting the trials for within corpus comparisons were inoperable in their original form for the program. In my implementation, I consistently completed my programs using a RandomForestClassifier from sklearn, rather than the Logistic Regression approach used in the original paper. This is because the run time for Logistic Regression as well as the data sizes were unworkable given my hardware.

In the end, very few of the F1 values that I received aligned with the F1 values reported in the original publication, and I was unable to replicate their results using their code.

## 4 Analyzing the Unified Corpus Paper

Given the improvements to automation of the trials that were absent from the original publication, which had code designed to be run manually through the terminal, it is unclear whether the results of the original paper are genuine, as it would have required them to have run the trials on code quite different from that which was published and to have had manual intervention for each trial, which, given that they claim to have used other classifiers, would have taken considerably longer than my application. The optimistic outlook on this paper would be simply that the code published on github was a flawed tool intended to help others without disclosing the original code. There is no conclusive judgement that can be made on the validity of the data or intentions of the original authors, other than to say that the published code accompanying the Unified Corpus paper is incapable of replicating their results.

Another issue encountered in the analysis of this paper was that of calculating corpus similarity. As previously mentioned, jensen shannon distance and chi square were initially intended to be tested, but through more thorough background research, I found that Chi Square would have already been shown to be a poor metric for small corpus comparison, such as this one. And for jensen shannon, I found that my numpy arrays were incompatible with the expected format for jensen shannon distance. Cosine similarity was fully implemented and verified on individual trials, however, it was found that there was a small, but cumulatively significant



memory leak when conducting consecutive trials. In runs of my application, it was found that for the final trials, corpus similarities were consistently returning values close to 1. I was able to find the mistake and fix it for cosine similarity comparison, but unfortunately, I was unable to find the error in per emotion cosine similarity, a metric I had hoped to use.

## 5 Conclusion

Through analysis of the Unified Corpus and the background research on generalizable approaches to emotion detection, it would seem that there remains much work to be done in terms of linguistic variation predictions. Current state-of-the-art methods such as ensemble methods that use LDA topic modelling to train classifiers off of topic features have shown promise in leveraging specialized corpus performance on general tasks. Techniques for corpus similarity leave room to be desired, but are improving, with the current state of the art method generally being the Jensen Shannon distance variants. Due to the necessity of some form of data isolation, either through manual annotation, or through topic modelling algorithms, research in generalizing emotion detection may continue to progress at a slow rate, as the field relies heavily on inter-researcher cooperation and publicly sharing high quality datasets. Given the ability to run larger models on better hardware in shorter timeframes, an area of improvement would likely be to use word embeddings for datasets of very consistent linguistic variations and to use these embeddings to train an initial classifier, similar to what is done with current state-of-the-art LDA ensemble models.

## Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

## References

Laura-Ana-Maria Bostan and Roman Klinger. 2018. [An analysis of annotated corpora for emotion classification in text](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

## A Appendices

Appendices are material that can be read, and include lemmas, formulas, proofs, and tables that are not critical to the reading and understanding of the paper. Appendices should be **uploaded as supplementary material** when submitting the paper for review. Upon acceptance, the appendices come after the references, as shown here.

**L<sup>A</sup>T<sub>E</sub>X-specific details:** Use `\appendix` before any appendix section to switch the section numbering over to letters.

## B Supplemental Material

Submissions may include non-readable supplementary material used in the work and described in the paper. Any accompanying software and/or data should include licenses and documentation of research review as appropriate. Supplementary material may report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.

Nonetheless, supplementary material should be supplementary (rather than central) to the paper. **Submissions that misuse the supplementary material may be rejected without review.** Supplementary material may include explanations or details of proofs or derivations that do not fit into the paper, lists of features or feature templates, sample inputs and outputs for a system, pseudo-code or source code, and data. (Source code and data should be separate uploads, rather than part of the paper).

The paper should not rely on the supplementary material: while the paper may refer to and cite the supplementary material and the supplementary material will be available to the reviewers, they will not be asked to review the supplementary material.