A graphical framework to detect and categorize diverse opinions from online news

Ankan Mullick
Department of Computer Science and Engineering
IIT Kharagpur, India

Pawan Goyal
Department of Computer Science and Engineering
IIT Kharagpur, India

Niloy Ganguly
Department of Computer Science and Engineering
IIT Kharagpur, India

{ankan, pawang, niloy}@cse.iitkgp.ernet.in

Abstract

This paper proposes a framework to extract diverse opinionated sentences within a given news article, by introducing the concept of diversity in a graphical model for opinion detection. We conduct extensive evaluation and find that the proposed modification leads to impressive improvements in performance and makes the final results of the model more usable. The proposed method (OP-D) not only performs much better than the other techniques used for opinion detection and introducing diversity, but is also able to select opinions from different categories (Asher et al., 2009). By developing a classification model which categorizes the identified sentences into various opinion categories, we find that OP-D is able to push opinions from different categories uniformly among the top opinions.

1 Introduction

Online publishing houses desire to develop engagement of users around the articles published on their websites. An important aspect of user engagement is commenting on the article and subsequently building up a conversation around it. In order to facilitate meaningful conversation, an option might be to identify and highlight *specific relevant portions* of the article, which may act as a seed for such conversation. For ensuring wide engagement, it would be best if the sentences chosen are opinions expressed in the article, as unlike factual statements, opinions might easily kick-start discussions. Further, to be able to engage a wide range of audience, it would be helpful if each chosen sentence expresses different context than the other. In general, all opinions are not of the same type. Opinions can be categorized into various categories and sub-categories (Asher et al., 2009), and it would be ideal if the extracted opinions cover multiple such categories. Some examples of these categories are provided below:

1) **Report :** e.g., *Christie's staffs have denied Zimmer's allegation.*

2) **Judgment :** e.g., *McGreevey's lover was being paid 11000 Dollar even though he was wildly unqualified for the position.*

3) **Advise :** e.g., *Let's shoot at the opposition not our own troops, one Insider pleaded.*

4) **Sentimental :** e.g., *So why do so many people enjoy ridiculing my New Jersey One word Jealousy?*

Opinion analysis has been a major field of study in natural language processing and data mining for many years. Several works such as (Kim and Hovy, 2006; Qadir, 2009; Scholz and Conrad, 2013; Yu and Hatzivassiloglou, 2003) focus on opinion mining. Opinion mining is very similar to subjectivity classification where subjective nature indicates the tendency of expressing one's thoughts and opinions. Work has been done in the past for developing classifiers (Wiebe and Riloff, 2005) which separate subjective sentences from objective ones, using several features present in the sentences. (Soni et al., 2014) describes how to predict certainty (factuality) of text (e.g., tweet) by using keywords collected from source introducing predicates (cues) and groups (Saurí, 2008). These models focus only on the local context (takes no global context into account) from a sentence to measure its subjectivity. Side by side, there have been works where graphical models have been proposed to capture the global context, where the sentences are treated as nodes in the graph and a similarity measure between sentences is defined to

Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, pages 40–49, Osaka, Japan, December 12 2016.

build this graph. While some of the approaches use PageRank to model each node in a similar manner (Erkan and Radev, 2004), HITS framework has also been used that establishes a relationship between opinions and supporting facts by modeling opinions as hubs and facts as authorities (Rajkumar et al., 2014).

None of these works, however, focus on finding diverse opinions from an article. Experiments on MPQA and Yahoo datasets (both are English datasets) show that this leads to a sub-optimal performance, while trying to extract the most opinionated sentences in a news article. The graphical models end up choosing similar sentences - which is not ideal to enable wide-ranging user engagement. We, therefore, attempt to modify a variant of graphical model proposed in (Rajkumar et al., 2014) to introduce diversity. The basic idea of our approach is that once a node (sentence) is selected as an opinion because of a high hub score (as per the HITS framework used in (Rajkumar et al., 2014)), we can discount the hub scores for the nodes it links to, as these might be sub-opinions supporting the main opinion, and discounting their hub scores might improve diversity. We find that this simple modification to the earlier framework leads to impressive improvement in performance for (i) Classifying opinions and facts, and (ii) Identifying diverse opinion categories in both datasets. Extensive experimental results are reported to show that as a result of this modification, the output opinions can be used more meaningfully. Note that the proposed technique is unique from the general work on diversity (Carbonell and Goldstein, 1998; Munson et al., 2009; Zhu et al., 2007; Mei et al., 2010) in that we introduce diversity in a model with two different kinds of nodes in the document graph, as opposed to the other algorithms, which treat all the sentences equally.

Further, to understand the distribution of extracted opinions from online news in various categories, e.g., Report, Judgment, Advise and Sentiment etc. (Asher et al., 2009), we develop an opinion classification model. While classification of opinions into various sentiment levels such as positive, negative and neutral has been tried (Saggionα and Funk, 2010; Yu et al., 2008), automated classification of opinions into various categories is not available. Analysis using the opinion classification model shows that our algorithm (OP-D) actually adds diversity even at the category level by selecting opinions from different opinion categories.

2 Extracting Diverse Opinions: OP-D

The proposed algorithm for extracting diverse opinions from news articles comprises of three steps: (i) Extracting features and assigning a score to indicate opinionatedness of a sentence. (ii) Building up the fact-opinion graph, applying HITS algorithm and identifying highly opinionated sentences. (iii) Identifying diverse opinionated sentences (i.e. report, judgment, advise, sentiment).

(i) Feature Extraction: We extract an extensive set of features at the sentence level to classify a sentence as an opinion / fact using a binary Naïve Bayes (NB) classifier. The features used include: (a) count of the strong polar words, weak polar words in the sentence (Wiebe et al., 1999), (b) polarity of the root verb of the sentence, (c) presence of *aComp*, *xComp* and *advMod* dependencies (Qadir, 2009), (d) opinionated n-grams (Wiebe et al.,), (e) presence of modal verbs, (f) presence of pronouns, (g) opinionated words (e.g., 'should', 'always', 'anyone', 'if' etc.). From LIWC (Pennebaker et al., 2001) we collected words belonging to the categories - 'feel', 'swear', 'certain', 'percept', 'time', as their presence in the sentence can make it more subjective or objective. A list of positive and negative polar words was used from MPQA opinion lexicon. Stanford dependency parser (De Marneffe et al., 2006) was utilized to compute the dependencies for each sentence within the news article. After those features are extracted from sentences, the Weka implementation[1] of the Naïve Bayes classifier is used to calculate the probability for each sentence to be an opinion, based on the presence of the above features.

(ii) Graph Formation and Hub-Authority Calculation: In this step, a graph is generated considering each sentence as a node. The scores from the NB classifier are used to assign the initial hub scores to the sentences in the graph. In HITS, edges flow from Hubs to Authorities, so an edge between two nodes (S_i to S_j) is given a higher weight W_{ij} if the source sentence has a high probability of being an opinion (probability value obtained from the NB classifier, which is also used to initialize the hub score of this

[1] http://www.cs.waikato.ac.nz/ml/weka/

sentence as $H_i(0)$ and Initial Authority-score as 1- $H_i(0)$.), the cosine similarity between these sentences (Sim_{ij}) is high and the number of sentences separating i and j, ($dist_{ij}$) is small. The weight function, W_{ij} [2] is as following

$$W_{ij} = {H_i}^3(0) \cdot Sim_{ij} \cdot (0.2 + \frac{1}{dist_{ij}}) \quad (1)$$

We now consider only the top $k\%$ of the edges, having the highest weights in the graph.

Once we have established a hub-authority structure in the document, we compute the hub and authority scores of every node in the graph by applying the HITS algorithm Ideally, the 'important' opinion sentences would obtain high hub scores because such sentences usually are stressed more in the article by using supporting facts *or* other related sentences; this results in high number of outgoing edges from these opinion sentences. Sentences supporting these opinions, similarly, should get high authority score.

Algorithm 1 OP-D Algorithm (output: k sentences)

```
procedure OP-D
    for All sentences do                                          ▷ Initialization
        hubscore ← value_ by_ NB_ classifier
        authscore ← 1 −hubscore
        Take top k% edges with edge weight 1 (thresholded). Set other edge weights to 0.
    end for
    while Root Mean Squared Error < ϵ do                          ▷ ϵ was set to 0.0001
        Update hubscore(h) & authscore(a)
        h ← AA^T h ; a ← A^T Aa                                   ▷ A:Adjacency Matrix
    end while
    while NoOfSentencesChosen < k do
        Sort the hub_scores and select the max
        Rank this maximum in final list
        Decrease the hub_score of the authorities of max_hub
    end while
end procedure
```

(iii). Ensuring Diversity: To introduce the notion of diversity in this framework, let us assume that we have selected a node i with the highest Hub score H_i as the opinion to be retrieved; and let node j be one of the authorities which has contributed to its hubness. Since we want the results to be more diverse, we decrease the hub scores of these nodes before selecting the next sentence with the highest hub score. The hub score of the authority (node j) is decremented by a fraction of the edge weight from hub (node i) to authority (node j), i.e.,

$$H_j \leftarrow H_j - \lambda.W_{ij} \quad (2)$$

where λ is a constant, H_j is the hub score of the authority node j and W_{ij} is the weight of the edge from node (hub) i to node (authority) j. The decrement ensures that these sentences do not get selected immediately when picking up the node with the next highest hub score. This process is repeated until we have selected the required number of opinionated sentences from the document. Steps are shown in **Algorithm 1**- *Opinion Diversity* (**OP-D**).

3 Dataset

To investigate the effectiveness of the proposed framework, experiments are conducted using two different datasets, a) the standard Multi-Perspective Question Answering (MPQA) dataset (contains 535 documents) and b) 120 news articles crawled from Yahoo news. Each document is a news article pertaining to some topic. In the MPQA dataset, each sentence is classified as either opinionated or factual by checking for the presence of certain subjective elements as annotated by the authors of the corpus (Wiebe

[2] In our approach W_{ij} is defined using parameters (Similar to (Rajkumar et al., 2014))

Table 1: Statistics of the MPQA and Yahoo datasets

Used	No.of Documents	Average length (no.of sentences) of an article	Average fraction of opinion sentences/article
Dataset	MPQA / Yahoo	MPQA / Yahoo	MPQA / Yahoo
Total	535 / 120	20.8 / 31	0.486 / 0.527
Train	435 / 95	20.2 / 31.4	0.49 / 0.524
Test	100 / 25	23.1 / 27	0.48 / 0.537

et al., 2005). For the Yahoo dataset, we get each sentence annotated manually using volunteers, different from the authors[3]. Statistics of these datasets, including the training-test splits, are provided in **Table 1.**

4 Experimental Framework and Results

From the 535 documents in the MPQA dataset, we randomly select 100 documents for the test set. Similarly, 25 documents are selected randomly from the Yahoo dataset for testing. Note that training and test splits are required for the first stage NB classifier. We first perform 5-fold cross validation experiments on the training sets. Then the entire training set is used to train the NB classifier and the results are reported on the test set.

Table 2: Variation of precision and recall for different k (% of edges) and weight (w) for OP-D

k(%)	(P,R)@3 wt=w	(P,R)@5 wt=w	(P,R)@3 **w=1/0**	(P,R)@5 **w=1/0**	(P,R)@3 wt=w	(P,R)@5 wt=w	(P,R)@3 **w=1/0**	(P,R)@5 **w=1/0**
	MPQA	MPQA	MPQA	MPQA	Yahoo	Yahoo	Yahoo	Yahoo
5%	0.6, 0.23	0.61, 0.37	0.64, 0.23	0.61, 0.39	0.72, 0.22	0.64, 0.33	0.75, 0.24	0.65, 0.34
10%	0.66, 0.27	0.63, 0.42	**0.73, 0.32**	**0.69, 0.45**	0.75, 0.25	0.68, 0.34	**0.79, 0.26**	**0.7, 0.36**
20%	0.63, 0.25	0.61, 0.38	0.66, 0.26	0.64, 0.4	0.7, 0.22	0.66, 0.34	0.72, 0.23	0.67, 0.34
30%	0.6, 0.23	0.58, 0.39	0.62, 0.24	0.61, 0.4	0.64, 0.18	0.61, 0.31	0.65, 0.19	0.63, 0.31
40%	0.59, 0.23	0.58, 0.39	0.61, 0.24	0.59, 0.4	0.59, 0.17	0.58, 0.17	0.61, 0.19	0.60, 0.29

Parameter Fixing: The parameters that we needed to fix for the proposed algorithm were: weight of the edges (absolute (wt) or thresholded (1/0)), k for the top $k\%$ edges if thresholded weights are used and the constant λ in Equation 2. **Table 2** shows results obtained by OP-D for different k and weight. In all the cases, we find that we get better results when the weight is properly thresholded rather than taking the absolute weight - we follow that henceforth. $k = 10$ performs the best, so we take top 10% of the edges. **Figure 1** shows how the precision and recall values vary for different choices of λ using 5-fold cross validation for all the experiments reported. Since the best results are obtained for $\lambda = 20$, that has been fixed for all the experiments.

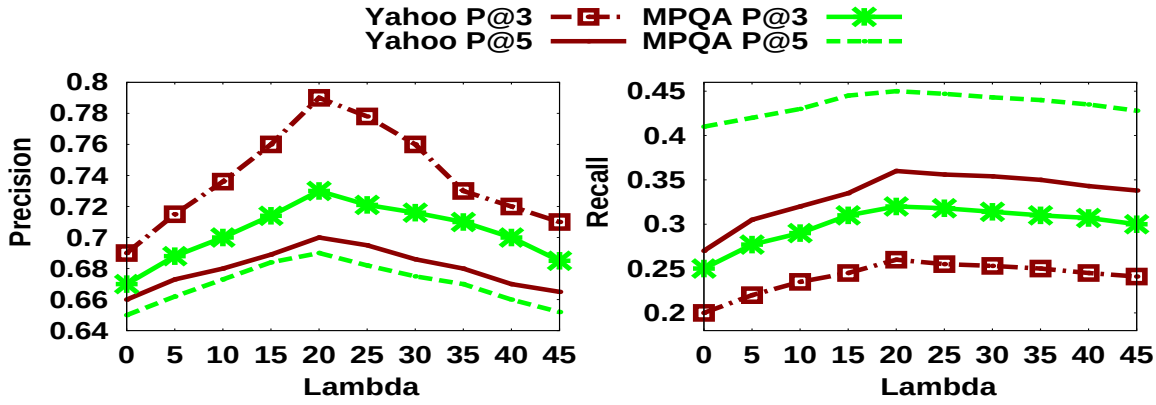

Figure 1: Variation of precision (P@3 and P@5) and recall (R@3 and R@5) for different values of λ.

[3]90 of these articles were obtained from Rajkumar et al. (2014). We increased this set to 120 articles. Inter-annotator agreement (Cohen κ) is 0.71.

Baselines: We use the following baselines for comparison:
a). **Random Baseline:** Each sentence is randomly assigned to one of the two classes, opinion or fact[4]. red We choose 3 and 5 sentences randomly to evaluate P@3 and P@5.
b). **Naïve Bayes (NB):** The next baseline is the first stage sentence-level classifier, as described earlier. We also experimented with Logistic Regression, SMO (Sequential Minimal Optimization), Support Vector Machine (SVM), Repeated Incremental Pruning - Java version (JRip) as well as other classifiers, however NB gave the best results and was used as a baseline.

c). **HITS (Rajkumar et al., 2014):** We use the method proposed by (Rajkumar et al., 2014) as another baseline. Note that the first two stages of our approach are similar to (Rajkumar et al., 2014) except that we use a more extensive set of features as well as the network is unweighted. We verified that these modifications indeed lead to performance gain, and used the modified approach as the baseline.
d). **FactJudge (Soni et al., 2014)**: FactJudge proposes a method to detect factuality of tweet. We use the factuality score given by this approach as a baseline to detect opinions, i.e., higher the factuality score, lower is the probability of being an opinion.
Evaluation Metrics: We use precisions $P@3$, $P@5$ and recalls $R@3$, $R@5$ as the evaluation measures, as we would like to obtain the best 3 or 5 opinions from the document. We perform a series of evaluations to compare the proposed approach with other baselines. In the first evaluation, we verify if the top 3 or 5 sentences returned by the algorithm are actually opinions. We also report the recall at top 3 or 5 places. We consider only those files (86 out of 100 for MPQA and 23 out of 25 for Yahoo) for testing which have at least one opinion.

Table 3: Comparison results of the proposed OP-D framework with other baselines on MPQA and Yahoo datasets

Number Of Article	Method Dataset	$(P, R)@3$ (MPQA)	$(P, R)@5$ (MPQA)	$(P, R)@3$ (Yahoo)	$(P, R)@5$ (Yahoo)
5-fold cross validation test on Training data	Random	0.5, 0.19	0.53, 0.3	0.54, 0.16	0.55, 0.26
	FactJudge	0.57, 0.19	0.56, 0.25	0.57, 0.15	0.57, 0.24
	NB	0.64, 0.28	0.63, 0.4	0.63, 0.17	0.62, 0.28
	HITS	0.67, 0.25	0.65, 0.41	0.69, 0.2	0.66, 0.27
	OP-D	**0.73, 0.32**	**0.69, 0.45**	**0.79, 0.26**	**0.7, 0.36**
Testing On: MPQA (86/100) ; Yahoo (23/25)	Random	0.52, 0.2	0.54, 0.32	0.53, 0.16	0.57, 0.27
	FactJudge	0.52, 0.19	0.53, 0.33	0.54, 0.15	0.54, 0.27
	NB	0.62, 0.27	0.61, 0.4	0.65, 0.17	0.63, 0.29
	HITS	0.66, 0.27	0.63, 0.4	0.69, 0.21	0.65, 0.32
	OP-D	**0.72, 0.31**	**0.68, 0.44**	**0.81, 0.26**	**0.71, 0.38**

Performance: **Table 3** shows the comparison results for the two datasets. We see that the random baseline, along with (Soni et al., 2014) gives a precision close to 0.5. The first stage NB classifier performs better than these methods. The graphical framework (second stage) gives further improvements upon the NB classifier, and OP-D outperforms all these baselines consistently at least by 5%.
Performance on different buckets of opinion fraction: Since the fraction of opinions in each document varies, we wanted to investigate the performance at various sparsity levels and thus study the robustness of the proposed algorithm. The test datasets were divided into various buckets according to the fraction of opinionated sentence (sparse, medium and dense) in the document. The results shown in **Table 4** confirm that OP-D performs better consistently across various buckets. Specifically, even for the documents with small fraction of opinions, it is able to improve performance from the NB and HITS baselines. In general, for the documents with sparse opinions, the performance is poor (across methods) which is bringing down the overall performance. This needs detailed future inspection.
Diversity Experiment: While these evaluations establish that the top 3-5 sentences extracted by OP-D contain more opinions than the baselines, they do not provide insights into whether these selected opinions are more important and diverse topics with respect to the entire article. We, therefore, randomly select 50 MPQA articles and 25 Yahoo articles, provide all the sentences with gold standard opinion /

[4]Python "random" module has been used.

Table 4: Comparison results of precision and recall on different buckets of opinion fractions

Opinion Faction (Doc/Total)	Method	$(P,R)@3$ MPQA	$(P,R)@5$ MPQA	$(P,R)@3$ Yahoo	$(P,R)@5$ Yahoo
(0-0.3)(22/86)	Random	0.15, 0.28	0.19, 0.42	0.33, 0.23	0.31, 0.31
For MPQA;	FactJudge	0.17, 0.24	0.18, 0.43	0.34, 0.19	0.34, 0.39
	NB	0.26, 0.43	0.22, 0.64	0.43, 0.2	0.41, 0.4
(0-0.5)(7/23)	HITS	0.27, 0.46	0.22, 0.59	0.62, 0.33	0.56, 0.51
For Yahoo	**OP-D**	**0.41, 0.53**	**0.35, 0.72**	**0.67, 0.41**	**0.57, 0.58**
(0.3-0.65)(31/86)	Random	0.47, 0.18	0.5, 0.31	0.41, 0.08	0.53, 0.17
For MPQA;	FactJudge	0.45, 0.19	0.47, 0.32	0.54, 0.09	0.5, 0.14
	NB	0.54, 0.25	0.55, 0.38	0.625, 0.12	0.55, 0.17
(0.5-0.65)(8/23)	HITS	0.66, 0.25	0.6, 0.39	0.7, 0.14	0.6, 0.19
for Yahoo	**OP-D**	**0.72, 0.28**	**0.64, 0.41**	**0.75, 0.15**	**0.68, 0.22**
(0.65-1)(33/86)	Random	0.81, 0.16	0.84, 0.27	0.77, 0.19	0.8, 0.32
For MPQA;	FactJudge	0.81, 0.16	0.82, 0.27	0.56, 0.15	0.63, 0.27
	NB	0.92, 0.18	0.93, 0.3	0.81, 0.19	0.8, 0.3
(0.65-1)(9/23)	HITS	0.91, 0.18	0.91, 0.29	0.75, 0.18	0.76, 0.3
For Yahoo	**OP-D**	**0.93, 0.19**	**0.94, 0.31**	**0.96, 0.25**	**0.85, 0.34**

fact labels to the annotators, and ask them to label 5 opinions, which they feel are important as well as diverse topics to cover the entire article. Each article is provided to 3 annotators and we use a rank aggregation method to prepare a gold standard of 5 *important* and *diverse* opinions from these articles.

Table 5: Comparison results for the most diverse set of opinions

Method	$P@3(50)$ MPQA	$P@5(50)$ MPQA	$P@3(25)$ Yahoo	$P@5(25)$ Yahoo
NB	0.322	0.344	0.35	0.31
HITS	0.41	0.4	0.42	0.4
MMR(On NB)	0.387	0.36	0.41	0.38
MMR(On HITS)	0.465	0.44	0.48	0.46
Grasshopper(On NB)	0.384	0.38	0.39	0.37
Grasshopper(On HITS)	0.471	0.44	0.493	0.48
DivRank	0.485	0.473	0.51	0.49
OP-D	**0.584**	**0.571**	**0.63**	**0.61**

We now measure $P@3$ and $P@5$ (Total annotated important and diverse opinions per article is 5, so recall is a simple function of precision, therefore we omitted.) depending on what fraction of the top 3 or 5 sentences returned by various systems feature in the 5 *important* and *diverse* opinions, as selected by the annotators. We use the standard diversity algorithms, MMR (Carbonell and Goldstein, 1998) and Grasshopper (Zhu et al., 2007), both on the results of NB and HITS, as well as DivRank (Mei et al., 2010) as baseline algorithms for diversity. **Table 5** shows that OP-D outperforms other methods (sometimes even by **10%**) in detecting diverse opinions. While both MMR and Grasshopper are able to achieve improvement over both NB and HITS classifiers, the order of improvement by OP-D over HITS is much higher, indicating that decreasing the hub scores of the authority of the selected hubs results eventually in more diverse opinions getting selected.

A goodness test of algorithms would be if the chosen sentences fall uniformly under various categories and sub-categories - this may instill diverse type of user engagement. We took the top 5 sentences for 23 Articles from Yahoo Dataset detected by OP-D, DivRank, Grasshopper (on NB), Grasshopper (on HITS), MMR (on NB), MMR (on HITS) and then got each sentence (opinions) labeled by 2 anonymous human annotators for the category and subcategories it belongs to. Any tie has been settled by another annotator. **Figures 2** and **3** show the distribution of opinions detected by these algorithms into various categories and subcategories respectively[5]. Clearly, OP-D is able to select the opinionated sentences from various categories and subcategories much more uniformly than the other algorithms. OP-D achieves the highest Shannon entropy among all the baselines reinforcing that claim. The performance can be even better

[5]For the sake of space, Fig. 3 shows only the best 3 algorithms.

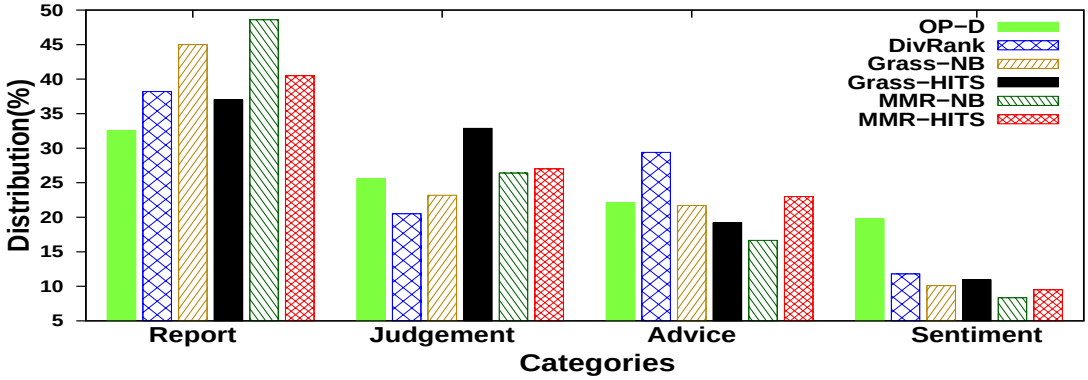

Figure 2: Distribution (Shannon Entropy) of opinions detected by OP-D (1.97), DivRank (1.88), Grasshopper on NB (1.82), Grasshopper on HITS (1.864), MMR on NB (1.73), MMR on HITS (1.8) into 4 broad categories.

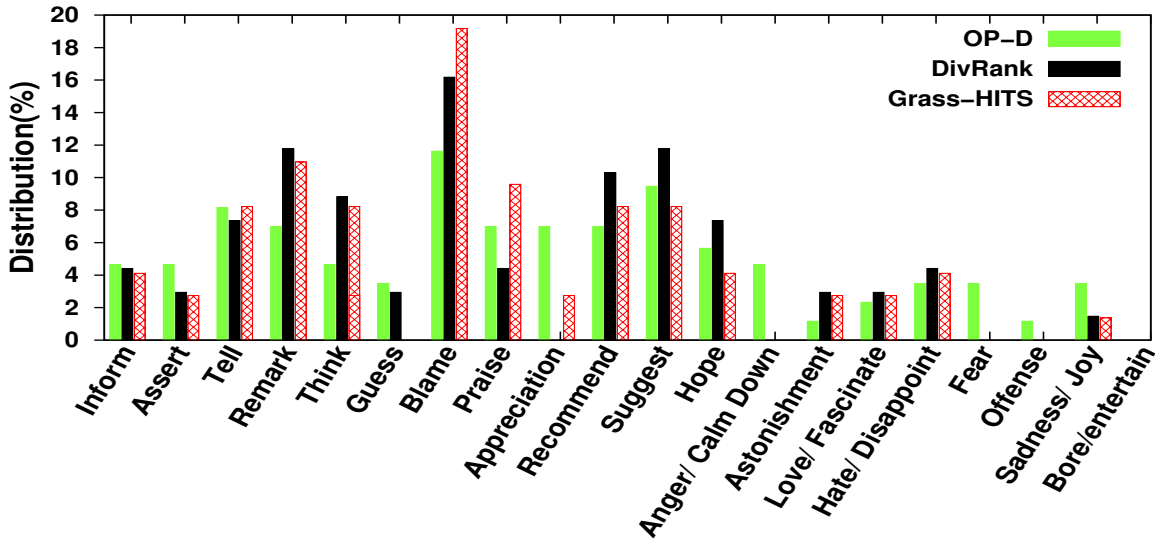

Figure 3: Distribution(Shannon Entropy) of opinions detected by best 3 methods- OP-D(4.05) DivRank(3.65),Grasshopper on HITS (S.E.= 3.63) into sub-categories.S.E. for Grasshopper on NB(3.54),MMR on NB(3.5),MMR on HITS(3.61).

appreciated if we can understand the overall distribution of opinions in an article. However, manually classifying all opinionated sentences is not possible - hence we build an automated classification model, which has been described in the next section.

5 Automatic Classification of Opinions into Opinion Categories

Dataset Preparation: 134 articles from MPQA and 29 articles from Yahoo dataset are taken randomly and have been annotated (with categories of opinion) manually using volunteers, different from the authors (Inter-annotator agreement Cohen κ is 0.8). Details of the datasets are provided in **Table 6**.

Table 6: Statistics of the Annotated MPQA and Yahoo dataset for automatic classification of opinion

Dataset	Sentences	Category	Sentences	Dataset	Sentences	Category	Sentences
MPQA	1237	Report	677	Yahoo	470	Report	216
		Judgment	247			Judgment	110
		Advise	132			Advise	79
		Sentiment	181			Sentiment	65

Features: We briefly describe the features used for automatic classification below:
(a) Sentence Length: Number of words in a sentence.

(b) Entropy of POS tags: After Parts of Speech (POS) tagging the sentence with Stanford POS tagger (De Marneffe et al., 2006), we take 14 POS tags (noun, pronoun, verb etc.) to calculate the Entropy of the probability distribution of POS tags in the sentence.

$$Entropy(i) = -\sum p_j * log_2(p_j) \tag{3}$$

(c) Positive, Negative and Neutral words: Number of positive, negative and neutral sentiment words. We checked it against standard positive, negative ((Rajkumar et al., 2014)) and neutral word set.
(d) Polarity of root verb: Polarity (+1, 0, -1) of the root verb is used as another feature.
(e) Average POS tag presence: We take average_word, average_letter_count, average_preposition, average_noun, average_pronoun, average_adjective, average_adverb for each category as features. For instance, average_noun for a category (e.g., reporting, advise etc) is computed as the average number of nouns per sentence of that category in the training set.
(f) Count of POS tags: Along with 5 different numeric features - count of noun, pronoun, adjective, adverb, preposition, we include 2 numeric features - count of weak adjectives and strong adjectives.
(g) Dependency Features: Count of adverbial clause modifier (advcl), adverb modifier (advmod), adjectival modifier (amod), clausal complement (ccomp), numeric modifier (num) dependencies are 5 numeric features (De Marneffe et al., 2006).
(h) Presence of Different Categories of Opinionated Words: From opinion groups and examples in (Asher et al., 2009), we collected words which are related to each of the four categories. Later we extended the wordset of each category by identifying similar words from wordnet (by calculating word similarity by path based approach) for Reporting, Judgment, Advice and Sentiment categories and created corresponding wordsets (4 binary features: 1 if word is present in the corresponding wordset, otherwise 0). Later, we manually checked every word in the wordset and removed words from the dataset which are not linked at all.
Classification Model: Initial datasets are imbalanced so we use SMOTE algorithm (Chawla et al., 2002) to make balanced datasets (w.r.t. number of reporting) and run several classifiers to obtain the best classification results. Repeated Incremental pruning - Java version (JRip), Logistic Regression (LR), Multi-Class Classifier (MCC), Naive Bayes (NB), Sequential Minimal Optimization (SMO), Support Vector Machine (SVM) available in Weka Toolkit (Hall et al., 2009) are used in the classification experiments.

Table 7: Comparison of 5-fold cross validation Accuracy (A), Precision (P), Recall (R), F1-Score (F) results for automatic classification of opinions for MPQA and Yahoo datasets.

	MPQA				Yahoo			
Method	A(%)	P	R	F	A(%)	P	R	F
JRip	**70.10**	**0.74**	**0.71**	**0.725**	**70.17**	**0.63**	**0.70**	**0.664**
LR	67.18	0.66	0.67	0.665	62.61	0.61	0.67	0.638
MCC	67.17	0.65	0.67	0.66	64.71	0.59	0.65	0.619
NB	53.68	0.51	0.54	0.525	50.42	0.55	0.51	0.53
SVM	53.12	0.53	0.53	0.53	64.3	0.5	0.64	0.561
SMO	65.73	0.68	0.66	0.67	68.48	0.59	0.68	0.632

Cross Validation: We first performed a 5-fold cross-validation using different classifiers. We achieve 70.1% accuracy, 0.74 precision (macro-average), 0.71 recall, 0.725 F-Score for MPQA and 70.17% accuracy, 0.63 precision (macro-average), 0.70 recall and 0.664 F-Score for Yahoo dataset for the task of opinion classification by the JRip classifier which produces better results than other five classifiers - Logistic Regression (LR), Multi-Class Classifier (MCC), Naive Bayes (NB), Sequential Minimal Optimization (SMO), Support Vector Machine (SVM). The results are shown in **Table 7.**

We now use this classifier to plot the opinion category distribution. We plot this distribution for top 3, 5 and 10 opinionated sentences retrieved by OP-D algorithm from the entire MPQA and Yahoo dataset. Then these top 3 (5 and 10) opinionated sentences are collected from each article into a set and the distribution of the set is plotted in **Figure 4**. We clearly observe that if we focus on the top 3 opinions

only, OP-D is able to select uniformly from the four categories. However, as we look at top 5 or top 10 opinions, more opinions from Report category come in, which might be due to the fact that Report category is more prevalent in the dataset.

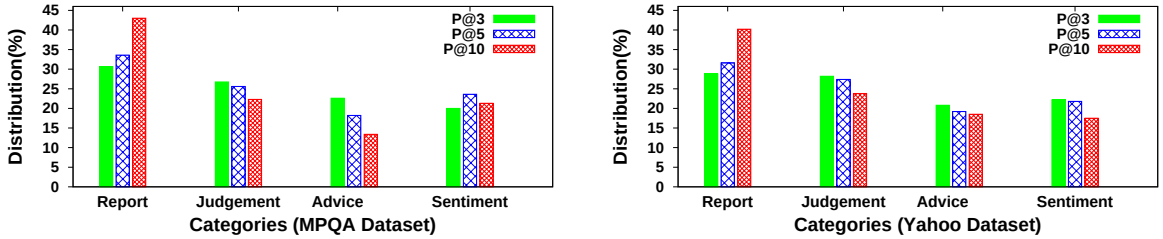

Figure 4: Distribution of opinion categories for top 3, 5 and 10 opinions retrieved by OP-D

6 Conclusion

In this paper, we first introduced diversity in a graphical framework to identify diverse and important opinions from a news article. Further, we built an automated classification model to classify the opinions into various opinion categories. Extensive evaluation establishes that the proposed modification helps in identifying the most diverse opinions from different opinion categories, giving a promising performance gain over the competing baselines. The top sentences returned by the algorithm can therefore be used to kick-start user discussions on a given news article. Building and deploying a system to that effect will be the immediate future step. Also, we would like to study more on how the distribution of opinions to facts, as well as across various opinion categories varies across various news categories.

References

Nicholas Asher, Farah Benamara, and Yvette Yannick Mathieu. 2009. Appraisal of opinion expressions in discourse. *Lingvisticæ Investigationes*, 32(2):279–292.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM.

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

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*, volume 6, pages 449–454.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Soo-Min Kim and Eduard Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. pages 1–8. ACL.

Qiaozhu Mei, Jian Guo, and Dragomir Radev. 2010. Divrank: the interplay of prestige and diversity in information networks. In *SIGKDD*, pages 1009–1018. Acm.

Sean A Munson, Daniel Xiaodan Zhou, and Paul Resnick. 2009. Sidelines: An algorithm for increasing diversity in news and opinion aggregators. In *ICWSM*.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001.

Ashequl Qadir. 2009. Detecting opinion sentences specific to product features in customer reviews using typed dependency relations. eETTs '09, pages 38–43.

Pujari Rajkumar, Swara Desai, Niloy Ganguly, and Pawan Goyal. 2014. A novel two-stage framework for extracting opinionated sentences from news articles. *TextGraphs-9*, pages 25–33.

Horacio Saggionα and Adam Funk. 2010. Interpreting sentiwordnet for opinion classification. In *Proceedings of the seventh conference on international language resources and evaluation LREC10*.

Roser Saurí. 2008. *A factuality profiler for eventualities in text*. ProQuest.

Thomas Scholz and Stefan Conrad. 2013. Opinion mining in newspaper articles by entropy-based word connections. In *EMNLP*, pages 1828–1839.

Sandeep Soni, Tanushree Mitra, Eric Gilbert, and Jacob Eisenstein. 2014. Modeling factuality judgments in social media text. *ACL*.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *CICLingLing*, pages 486–497. Springer.

Janyce Wiebe, Theresa Wilson, and Matthew Bell. Identifying collocations for recognizing opinions. In *ACL-2001 Workshop*, pages 24–31.

Janyce M Wiebe, Rebecca F Bruce, and Thomas P O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. pages 246–253. ACL.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. pages 129–136. EMNLP.

Bei Yu, Stefan Kaufmann, and Daniel Diermeier. 2008. Exploring the characteristics of opinion expressions for political opinion classification. In *Proceedings of the 2008 international conference on Digital government research*, pages 82–91. Digital Government Society of North America.

Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. Improving diversity in ranking using absorbing random walks. In *HLT-NAACL*, pages 97–104. Citeseer.