Product Review Classification from Twitter Data using Semisupervised Learning

Vinamra Singh

***Abstract*—Twitter and similar microblogging platforms have gained popularity in Latin America, fostering communities discussing various topics, including product experiences. As businesses aim to understand customer preferences, sentiment analysis and topic understanding become crucial. This paper focuses on classifying short-text comments from product reviews extracted from Twitter. Two semisupervised techniques, Label Propagation and a variation of Structural Learning, are employed to improve classification performance with limited labeled data. The challenges posed by short text and noisy data are addressed, and experimental results are presented, showcasing the effectiveness of the proposed methods compared to traditional supervised learning. Future directions for further improvements are also discussed.**

***Index Terms*—Semisupervised learning, Label Propagation, variation of Structural Learning, customer preferences, sentiment analysis , topic understanding.**

I. INTRODUCTION

TWitter and Tuenti are popular microblogging services in Latin America, as the prices of smartphones became more accessible to people, big communities have grown around these services. These communities are actively commenting on everything from political events to product experiences. Therefore, it has become of interest for many companies to analyze the microblogging data to understand what their clients want of their products, this led naturally to sentiment Analysis [1]

One of the subtasks of Sentiment Analysis is to find the topic or aspect of an opinion; for example, given a set of opinions about a mobile phone, it is possible to label each comment with the topic it refers to; these labels could be “battery”, “design”, “operative system,” etc. This Subtask can be modeled as a classification problem, given a set of fixed possible topics.

In the scenario of text classification, semisupervised techniques present two advantages: First, a semisupervised method should allow the classification of text by annotating only a small portion of data, thus reducing the labor of annotating. Second, the unlabelled data is used during training; this is important since the amount of unlabelled data in these tasks is abundant and available, therefore it can be used to provide extra knowledge to the model.

Zhang and Kubota [2] proposed a semisupervised approach using linear classifiers for multiple learning tasks. They proposed to create additional classification tasks (Auxiliary Problems) aside from the Target Problem. The underlying idea is that the auxiliary problems will help find good predictive structures [3].

One of the constraints in this approach is that auxiliary problems should be able to generate labeled data from the original unlabelled data automatically. This method has been used in tasks such as Text Chunking [4].

In contrast, [5] proposes a graph approach called label propagation for semisupervised learning. This approach maps the data to a graph representation, then labeled instances propagate their labels through the graph, allowing unlabelled data to adopt the label of similar instances. [6] uses Label Propagation for doing Polarity Classification on tweets.

Even though classifying text has been a widely studied topic, the focus has been on long documents, whereas tweets are at most 144 characters long [7]. The new trends in these social networks have led to research about classification in short texts; the latter has shown that it raises new challenges, and the former approaches are ineffective. One of the reasons is that user-generated comments in the mentioned services tend to be extremely short, leading to sparse feature representations [8].

Concerning short text classification, Xinghua and Hongge [9] propose to do Feature Extension to deal with data sparsity; in their approach, each comment is extended with extra words from an expansion vocabulary. On the other hand, [10] proposes to use encyclopedic knowledge from Wikipedia to extend the short comments. As opposed to the above ones, [11] reduces the word space of the comments to keywords and uses information retrieval with a voting scheme to find labels for short comments.

This paper describes the setup of an experiment for classifying short-text comments of product reviews extracted from Twitter. Two semisupervised techniques are used, Label Propagation [5] and a variation of Structural learning problem for multitasks [2]. The next sections describe briefly the preprocessing of the tweets, each of the proposed semisupervised methods, and how they were adapted to the task. Additionally, the results are shown, where both methods are compared among themselves and to a supervised approach.

In order to provide a more comprehensive understanding of the current research landscape in the field of sentiment analysis, it is crucial to delve deeper into prior relevant work. While the introduction briefly touches upon the importance of sentiment analysis, a more extensive review and analysis of existing literature can offer readers a more nuanced perspective [12]. This will not only establish the significance of the study within the broader context but also facilitate a more informed discussion of the proposed semisupervised learning techniques [13].

Vinamra Singh (e-mail: vinamrasingh19@gmail.com).

Manuscript received October 30, 2023, Revised February 10, 2024, Accepted February 18 , 2024

II. Datasets

The Datasets of this experiment were provided by Meridian. Meridean[1] is a Colombian company that extracts tweets mentioning Latin American companies or products.

The datasets are separated by product. There are 3 datasets in this experiment:

- Sportswear dataset
- Mobile-Phone dataset
- Hygienic-product dataset

Each dataset is composed of tuples; each tuple contains the source of the opinion, the opinion, the polarity, and the topic.

Comment: #ProbandoXperiaArcS Gracias @TalkMex. Las fotos se ven tan nitidas.
Translation: #TestingXperiaArcs Thanks to @TalkMex. Pictures are very sharp.
Topic: Camera

TABLE I: Sample Comment

Each dataset contains approximately 5000 tuples.

III. Pre-processing

The Datasets used in this experiment are actual data crawled from Twitter and other sources, therefore, there is a lot of noise in them. Some of the problems are (but not limited) to:

- Miss spelled words, from typo errors to lack of accents.
- Internet Language, replacing some letters for others whose phonetics are similar i.e.: "**qu**iero" (I want) for "**k**iero", abbreviations and expressions ("Jaja" "jiji" ...)
- Twitter Jargon such as: "RT:","@"..

The pre-processing done was the following:

1) Remove Strange characters, such as hearts and other Unicode characters
2) Remove some of the Twitter Jargon
3) Use http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/TreeTagger [14] for getting the part of speech and the stem of each word in each comment
4) discard those words that are not adjectives, nouns or verbs
5) replace each word by its stem in lowercase
6) remove accents from words

When it comes to pre-processing, there could be a wide number of possible tasks that can be done; I try to keep it simple given the time constraints [15].

As a result of this pre-processing, each comment is converted into a bag of keywords. This will be later transformed into a vector space representation.

IV. Label Propagation

This semisupervised learning graph method was proposed by Zhu [5]. The idea behind Label propagation is similar to K-Neighbours, nevertheless, Label Propagation makes use of the unlabelled data during the training process.

This approach maps the data to a graph representation. In this representation, each arc of the graph will connect two nodes only if the two nodes are similar; the weights of the arcs are directly proportional to the similarity of the incident nodes.

Concerning our classification task, each node in the graph corresponds to the feature vector of a comment. The cosine distance among the vectors gives the weights of the arcs. The final representation is a complete graph. [16]

In the LP algorithm, the label information of any node in a graph is propagated to nearby nodes through weighted arcs until a global stable stage is achieved [17]. The label will travel more easily through the graph if the weight arc is high.

The LP Algorithm iterates until convergence; at each step, the algorithm will push the labels of the labeled data points through the arcs of high weights. The underlying assumption is that instances among a class will have high-weighted arcs connecting them. At each iteration, the labels of the labeled data are clamped.

This propagation is done by measuring a transition matrix T. The transition matrix states the probability of propagating the label of an instance, and it is measured from the weights of the graph's arcs.

Let T_{ij} be the probability of propagating the label of instance i to instance j, then it can be defined as:

$$T_{ij} = P(i \to j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}} \tag{1}$$

A. Algorithm

Let Y_{ij} be the soft probability of labeling instance i with label j.

Given our semisupervised task, then Y can be divided into:

Y_U:the soft labels for the unlabelled data
Y_L:the soft labels for the labelled data

This means that Y_L is given at the beginning of the problem and the target is to find good values for Y_U.

Step 1 - Init: In this step, the labels for Y_L are clamped, and the labels for Y_U are randomly initialized.
The proof that the initialization values of Y_U are not transcendental can be found at [5].

Step 2 - Propagation: In this step, the labels of neighboring nodes are pushed.
Let t be the current iteration, then Y^{t+1} is defined as:

$$Y^{t+1} = TY^{t} \tag{2}$$

Step 3 - Clamp Labelled Data: In this step, the Labels for Y_L are clamped.

[1] http://merideangroup.com/

Step 4 - Repeat: Repeat from step 2 until convergence.

Step 5 - Labeling Approach: Once there is convergence, a label has to be chosen for each node in the graph; there is more than one approach.
The simplest approach would be to pick the label with the highest probability.
In [5], there is an empirical analysis on more sophisticated approaches, showing how this criterion can affect performance; nevertheless, the simple approach was used for this experiment.

More sophisticated versions of this approach have been presented and assessed in [18] Talukdar and Pereira showed an empirical comparison among different variations of the LP algorithm, a modified absorption algorithm is presented where seed labels are not clamped, and it is shown to have better performance in the given tasks. Nevertheless, within this paper, I call LP to the algorithm proposed by Zhu [5].

To summarize our experiment, I convert our data into a complete graph; each node is the feature vector of a comment, and the weight of each arc corresponds to the similarity of the comments being incident to the arc. This Graph is fed to the Label Propagation Algorithm proposed by Zhu.

V. STRUCTURE LEARNING

A semisupervised learning method proposed in [2]. This paper proposes a way to learn multiple classification tasks. This approach extends the original Classification Task(Target Problem) by creating a set of auxiliary classification tasks. The auxiliary classification tasks are used to improve performance on the target task; the underlying assumption is that by solving the auxiliary problems, one should be able to find good predictive structures. The Auxiliary Tasks are trained on the unlabelled data, in consequence, one constraint to the creation of auxiliary tasks is that they should be able to automatically generate training data from the target's problem unlabelled data. In [4], it is argued that having a high number of auxiliary tasks is beneficial for performance.

In this paper's experiment, the target task corresponds to finding the topic of each comment. The auxiliary problems are defined as: learning a linear predictor using the Unlabelled data for the K most common words in the unlabelled data and a predictor for the words with the highest Pointwise mutual information(PMI) from the labeled Data. By doing this, both the features of the labeled data and unlabelled data will be extended; the labeled data will be extended with features from the unlabelled data and the unlabelled data will be extended with predictive features of the labeled data.

For learning this task, the words with the highest PMI are first found using the labeled data, and then the words with the highest frequency in the unlabelled data are found. Subsequently, a linear classifier for predicting each of the most common words is trained using the unlabelled data, and a linear classifier for predicting each of the most predictive words is trained using the labeled data. Labels for these tasks can be easily generated by masking the words for which the classifiers are being trained to predict.

By solving the auxiliary tasks, the feature vectors of the training data for the target problem can be extended with the predictions of the auxiliary tasks. Also, the vectors of the unlabelled data are extended. This might be useful in the experiment setup since I am extending the feature representations from knowledge of unlabelled data.

Finally, a linear classifier is learned using the training data for the target problem. Concerning the ASO-SVD algorithm, this paper explores a naive approach by using auxiliary problems. In this regard, the auxiliary classifiers extend the feature vectors [19] For the rest of the paper, I will refer to this approach as Naive SL.

VI. EXPERIMENT

The three datasets mentioned above were used as training and test data for the experiments. The experiments correspond to the following:

- Single task Supervised Learning using SVM (SVM)
- Single task semisupervised Learning using Label Propagation (LP)
- Single task semisupervised Learning using The naive Structural Learning(SL)

Each of the experiments was ran with different amounts of training data, specifically 10%, 30%, 60%, and 90%. For all the linear classifiers trained in the experiments, a Polynomial Kernel was picked, and the value of $gamma$ was 1.2; the reason behind this choice is motivated by the empirical results obtained by [20] in a similar task.

For making the tasks suitable for the given computational power, a subset of the size of $\frac{1}{4}$ of each of the given datasets was used; this has a significant drawback since the more the data, the better the behavior of the semisupervised algorithms. Furthermore, an upper bound on the number of auxiliary classifiers created by the naive structure learning was imposed; the auxiliary problems would be 40% of the most common words in the unlabelled data and the words with the highest PMI from the Labelled data. A major drawback of Label Propagation from a running time perspective is calculating a complete graph containing the node similarity. Calculating a graph with 2 or more millions of arcs consumes lots of resources.

VII. RESULTS

Overall, the Naive SL could maintain the f-score given by the supervised approach and, in some cases, was able to get a much better f-score by improving recall; in some other cases, a slightly better recall resulted in a decline of the precision, thus leading to a worse f-score.

The LP approach on the other hand, raised in labels where both of the previous approaches failed, such as with the label $Resistance$ in the sportswear dataset Figure 4 or $Brand\ Support$ Figure 1, $Tenderness$ in the Hygienic Product dataset Figure 6, and many others like $Photography$,$Screen$ and $Social\ Networks$ in the Mobilephone dataset Figure 12.

The Hygienic Product and Sportswear datasets yielded decent results, whereas the Mobile Phone dataset was challenging.

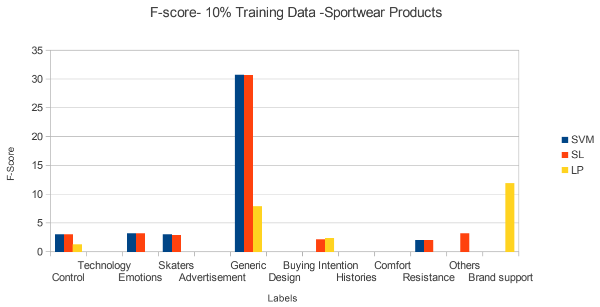

Fig. 1: F-Score, 10% Training Data, Sportwear dataset

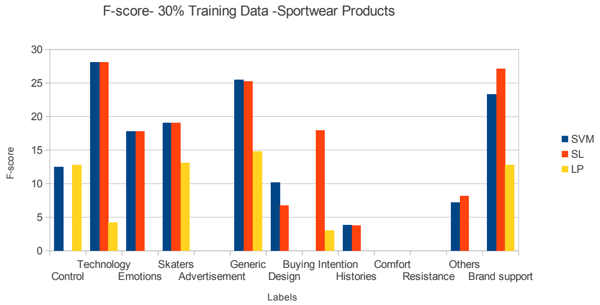

Fig. 2: F-Score, 30% Training Data, Sportwear dataset

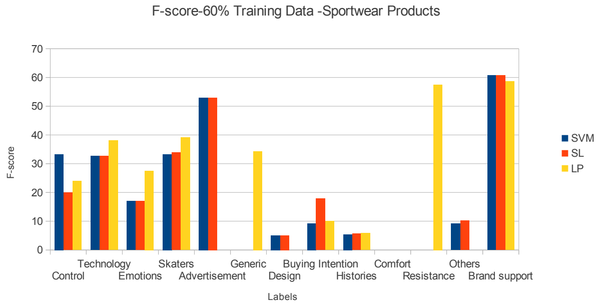

Fig. 3: F-Score, 60% Training Data, Sportwear dataset

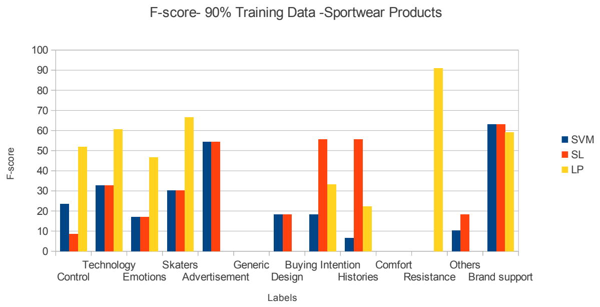

Fig. 4: F-Score, 90% Training Data, Sportwear dataset

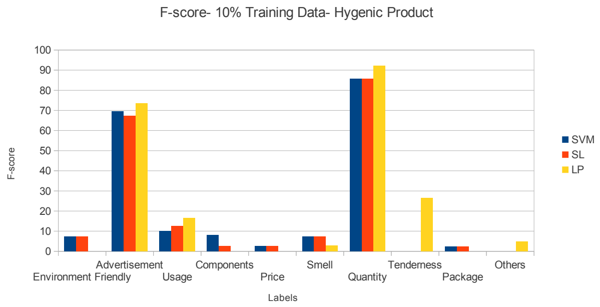

Fig. 5: F-Score, 10% Training Data,Hygienic Product

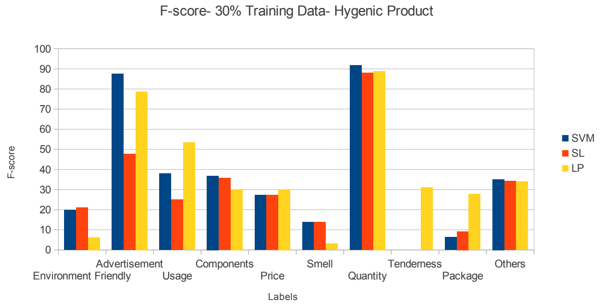

Fig. 6: F-Score, 30% Training Data, Hygienic Product

One factor that might have affected the overall performance was using a subset of the data. This especially seems to affect LP. LP profits from abundant unannotated data; I hypothesize that using a minimal set of seeds with many unannotated data should improve the scores below. The reason for LP giving such low scores on some of the tasks might be caused by the limitation on the dataset sizes and the data sparsity in the feature vectors.

First, the Limitation on the size of the datasets might have caused a lack of extra samples, thus generating some disconnected segments of the graph in the case of LP, therefore, proper propagation could not be carried out successfully for prediction.

Second, the feature vectors are very sparse because of the bag of keywords approach and the short text nature of the task, thus leading to disconnected segments of the graph.

A second test was carried out to discover whether more data and selecting more features would benefit the approach; this second test was done on the Mobile Phone dataset. In this trial, more data were taken into account, and a more significant number of features were allowed to be part of the feature vectors; this would also benefit SL since the number of auxiliary problems would be increased. The results are given in Figure 11 and Figure 12; it can be said that there is much a better f-score for a larger number of labels when compared to the previous tests, so It is possible to speculate that using the whole datasets would greatly increase the f-scores given in this report.

Since comments are reduced to a bag of keywords, other factors that could have caused the given results are:

- **Noise**: the given data is actually user-generated content

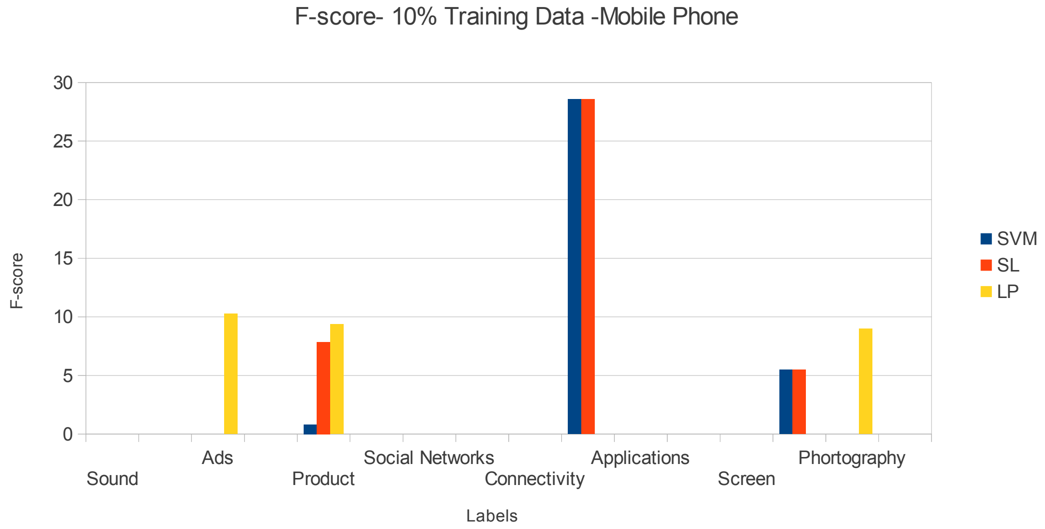

Fig. 7: F-Score, 10% Training Data, Mobile Phone dataset

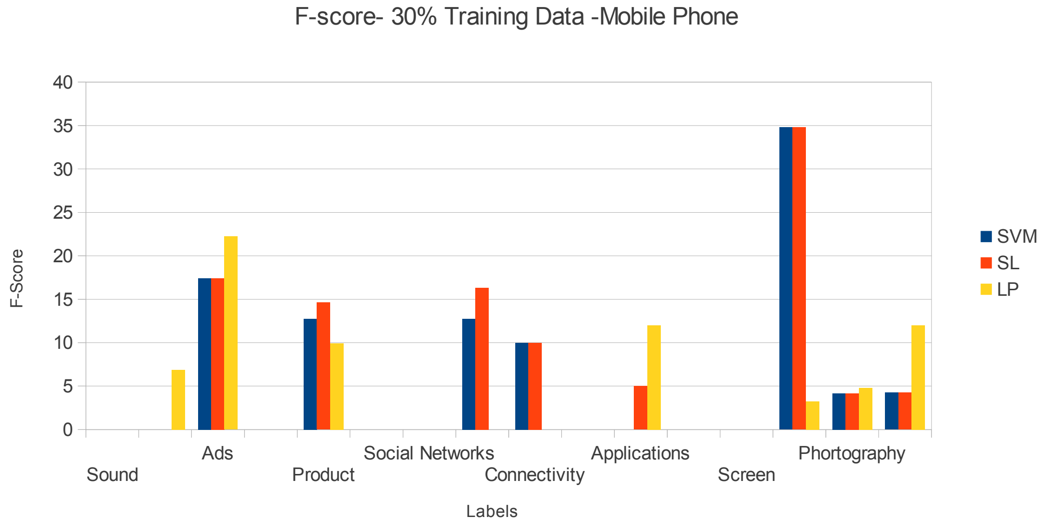

Fig. 8: F-Score, 30% Training Data, Mobile Phone dataset

and thus very noisy.

Noisy data, such as miss written words, can lead to bad classifications or points isolated in the LP graph. Take as an example the label $Camera$ for the Mobile phone dataset; if a good predictor for this label is the actual word “camera” and this word has been seen in the labeled data, seeing variations such as “camera” might not result predictive, since they would be in a different dimension.

- **Sparsity and Short Text**: feature vectors are very sparse, and text is short. This could affect in the following ways, Let’s think again about the label $Camera$ for the Mobile phone dataset; if I assume I am classifying long texts, an annotated instance can reveal a lot of the topic vocabulary, in this case, words such as “resolution”, “megapixels” and so on. On the other hand, in the short text classification, you got at most 144 characters per text so you might end up only with a small subset of topic-related words. For example, let’s assume you have a set of good predictors in your labeled data, which are the words: “Camera” and “resolution.”If there is a comment saying just: “it has a good amount of megapixels”, it will be doubtful to be classified correctly unless there is another comment linking “megapixels” with one of the predicting words in the seed set.

Concerning the topic vocabulary, I speculate that this really affects the performance. In the Hygienic Products and Sportswear, the term’s range is more or less reduced; this is not the case of the mobile phone dataset. The range of topic vocabulary for each label is enormous; take into account the label $Application$; a comment can

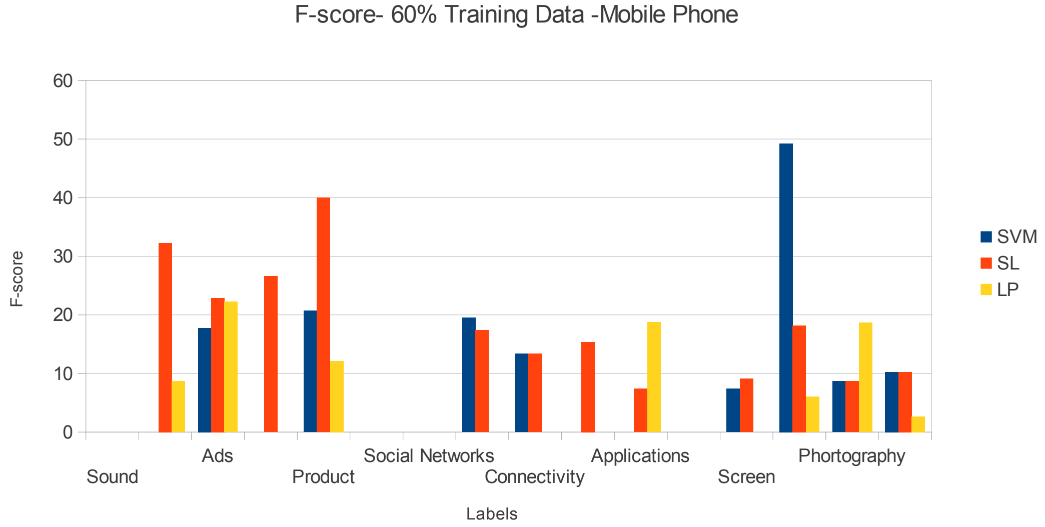

Fig. 9: F-Score, 60% Training Data, Mobile Phone dataset

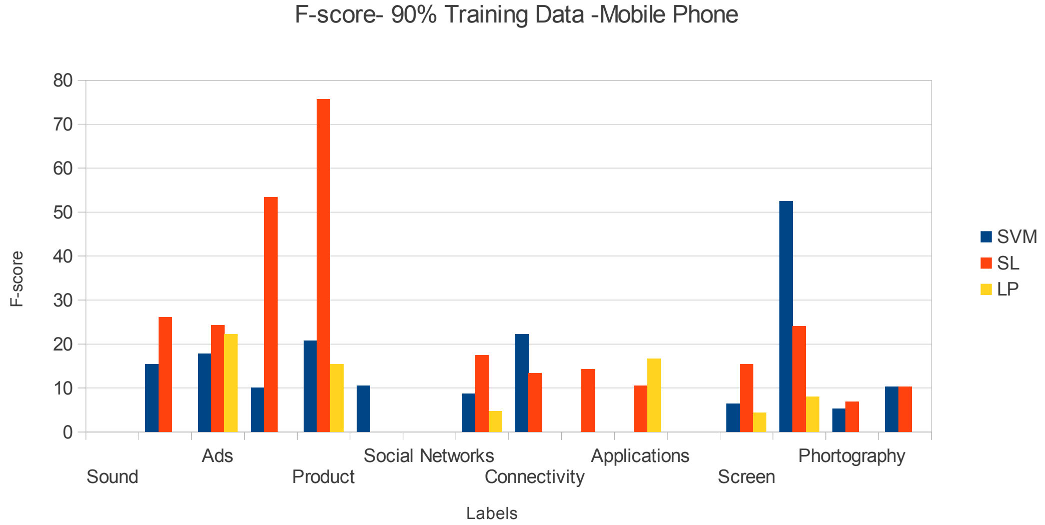

Fig. 10: F-Score, 90% Training Data, Mobile Phone dataset

be given to this label if it has a mention to a well-known IOS or Android App, such as "Angry Birds", take into account the label $Social\ Network$ where social networks' names are mentioned. On top of that, combine all the possible technical words that might appear with the labels $Camera$, OS, $Photography$.

VIII. CONCLUSIONS

An experiment on classifying short comments about products was described, two semisupervised approaches were presented and compared for solving this problem.

Data sparsity and the length of the texts were shown to be an issue for Label Propagation. One idea for tackling this problem would be injecting some knowledge related to the labels; this could be useful in difficult tasks such as the Mobile Phone dataset. An approach similar to the one given by [10] could be tried. Basically, Wikipedia could be used to gather Topic vocabulary and use that vocabulary to extend each of the comments; this could solve disconnected segments in the graph.

Another idea worth trying is to use the SL for multitask classification instead of tackling each dataset separately; joining them and solving them as a single classification problem might be beneficial. However, this requires more computational power, but at the same time, it should allow better results.

Despite noise being a problem, I would speculate that given enough data, the problem of noisy entries such as misspelled words should not be a big issue, assuming there is enough noisy data in both the labeled and unlabelled entries.

It would also be interesting to try the other Label Propaga-

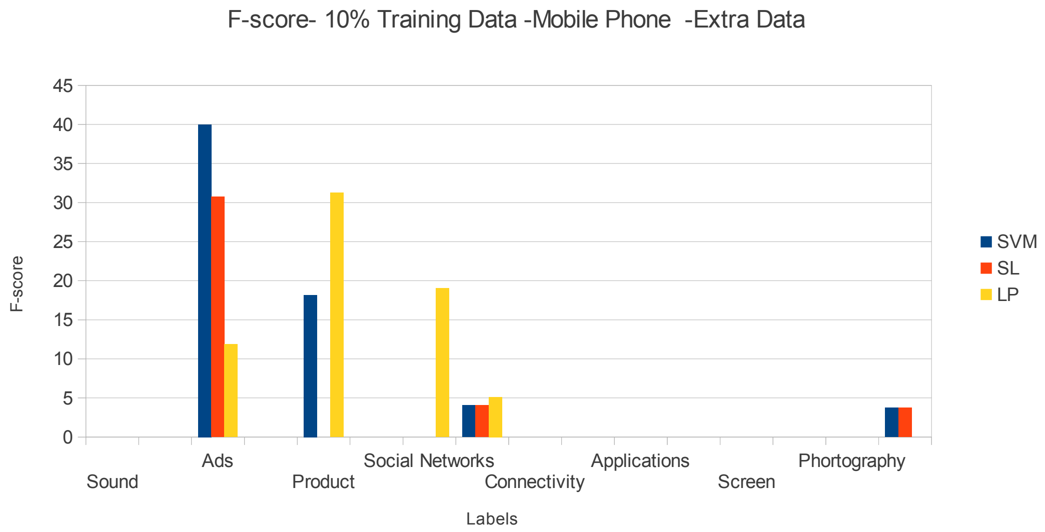

Fig. 11: F-Score, 10% Training Data, Mobile Phone dataset, More Features and Data

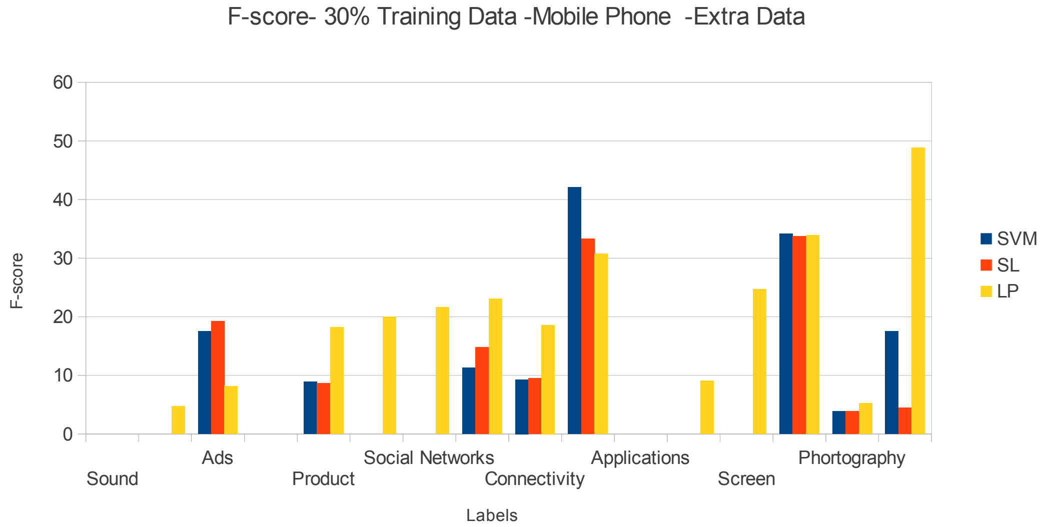

Fig. 12: F-Score, 30% Training Data, Mobile Phone dataset,More Features and Data

tion variations, such as Modified Adsorption (MAD) discussed in [18]; these implementations are available in Junto. MAD should help when dealing with noisy input labels, which was one of the cases between $Photography$ and $Camera$ in the Mobile Phone dataset.

Finally, joining the naive SL given here with LP should help the latter get better scores by allowing to connection some of the isolated sub-graphs generated by the feature sparsity.

In conclusion, a comprehensive exploration of the current research landscape in sentiment analysis is indispensable to provide readers with a holistic view of the field. Our literature review has delved into a diverse range of studies, from traditional methods to recent innovations, contributing to the ongoing dialogue in this dynamic and evolving domain. This thorough examination sets the stage for our study, which seeks to build upon existing knowledge and introduce novel semisupervised techniques to address the complexities of short-text sentiment analysis.

References

[1] P. Basile, V. Basile, M. Nissim, N. Novielli, V. Patti *et al.*, "Sentiment analysis of microblogging data." 2018.

[2] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Dec. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1046920.1194905

[3] H. Askr, E. Elgeldawi, H. Aboul Ella, Y. A. Elshaier, M. M. Gomaa, and A. E. Hassanien, "Deep learning in drug discovery: an integrative review and future challenges," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 5975–6037, 2023.

[4] R. K. Ando and T. Zhang, "A high-performance semi-supervised learning method for text chunking," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ser. ACL '05.

Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 1–9. [Online]. Available: http://dx.doi.org/10.3115/1219840.1219841

[5] X. Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, Pittsburgh, PA, USA, 2005, aAI3179046.

[6] M. Speriosu, S. Upadhyay, N. Sudan, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph."

[7] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy, "An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit," *Information Processing & Management*, vol. 57, no. 2, p. 102034, 2020.

[8] M. Yang, Y. Ren, and G. Adomavicius, "Understanding user-generated content and customer engagement on facebook business pages," *Information Systems Research*, vol. 30, no. 3, pp. 839–855, 2019.

[9] X. Fan and H. Hu, "A new model for chinese short-text classification considering feature extension," in *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence - Volume 02*, ser. AICI '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 7–11. [Online]. Available: http://dx.doi.org/10.1109/AICI.2010.125

[10] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge," in *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, ser. AAAI'06. AAAI Press, 2006, pp. 1301–1306. [Online]. Available: http://dl.acm.org/citation.cfm?id=1597348.1597395

[11] A. Sun, "Short text classification using very few words," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '12. New York, NY, USA: ACM, 2012, pp. 1145–1146. [Online]. Available: http://doi.acm.org/10.1145/2348283.2348511

[12] M. V. Mäntylä, D. Graziotin, and M. Kuutila, "The evolution of sentiment analysis—a review of research topics, venues, and top cited papers," *Computer Science Review*, vol. 27, pp. 16–32, 2018.

[13] X. Zhu and A. B. Goldberg, *Introduction to semi-supervised learning*. Springer Nature, 2022.

[14] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," 1994.

[15] A. Truong, A. Walters, J. Goodsitt, K. Hines, C. B. Bruss, and R. Farivar, "Towards automated machine learning: Evaluation and comparison of automl approaches and tools," in *2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI)*. IEEE, 2019, pp. 1471–1479.

[16] S. Dabhi and M. Parmar, "Nodenet: A graph regularised neural network for node classification," *arXiv preprint arXiv:2006.09022*, 2020.

[17] J. Chen, D. Ji, C. L. Tan, and Z. Niu, "Relation extraction using label propagation based semi-supervised learning," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ser. ACL-44. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 129–136. [Online]. Available: http://dx.doi.org/10.3115/1220175.1220192

[18] P. P. Talukdar and F. Pereira, "Experiments in graph-based semi-supervised learning methods for class-instance acquisition," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1473–1481. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858830

[19] P. A. Harr, A. Jordi, and L. Madaus, "Analysis of the future change in frequency of tropical cyclone-related impacts due to compound extreme events," in *Hurricane Risk in a Changing Climate*. Springer, 2022, pp. 87–120.

[20] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, ser. ECML '98. London, UK, UK: Springer-Verlag, 1998, pp. 137–142. [Online]. Available: http://dl.acm.org/citation.cfm?id=645326.649721

Vinamra Singh graduated from Amity University, majoring in Computer Science. His research interests lie in algorithmic development in the field of machine learning. He is keen to further advance his contributions by actively participating in pioneering developments within applied computing, with a steadfast commitment to advancing the fields of machine learning and natural language processing.