



# **Deliverable D 7.3: Algorithms and Evaluation Report**



## **Ubiquitous participation platform for Policy making**

### **UbiPOL**

**Brunel University  
Barnsley Metropolitan Council  
Turksat  
Basarsoft  
Corvinus University of Budapest  
Sabanci University  
PDM&FC  
Fraunhofer FOKUS  
IPASA**



**Project reference: ICT-2009-248010**

	<b>UBIPOL – ALGORITHM AND EVALUATION</b> Del. no: D7.3	
<b>ICT-2009-248010</b>	<b>Version: internally approved</b>	<b>14/04/2013</b>

**Deliverable type:** R

**Classification:** PU

**Work package and task:** WP7, T7.4

**Responsibility:** SU

**Executive summary:** This report has three parts:

First, we identify the privacy problem regarding public opinions and propose a new probabilistic privacy model MSA-diversity, specifically defined on datasets with multiple sensitive attributes. Then, we introduced formal definitions for the security and privacy requirements of keyword search on encrypted cloud data including hiding the search pattern. Also we proposed a scheme that uses cryptographic techniques as well as query and response randomization. Also we extend our opinion mining engine. We evaluated new features to be used in a word polarity based approach to sentiment classification and we considered different aspects of sentences, such as length, purity, unrealistic content, subjectivity, and position within the opinionated text. This analysis is then used to find sentences that may convey better information about the overall review polarity (Gezici et al., 2012). Later, we worked the effect of subjectivity-based features on sentiment classification on two lexicons and proposed new subjectivity-based features for sentiment classification (Dehkharghani et al., 2012) Finally, we addressed the problem of adapting a general purpose polarity lexicon to a specific domain and propose a simple yet effective adaptation algorithm (Demiroz et al., 2012).

---

**Amendment History**

Date	Issue	status	Author
25-11-2012		The first draft of the document by SU	Dilek TAPUCU Yücel SAYGIN
18-12-2012		Second version of the document	Dilek TAPUCU Yücel SAYGIN
24-03-2013		New section added: Implementation and Evaluation on UBI POL Data Sets	Dilek TAPUCU Yücel SAYGIN
11-04-2013		Corrections completed	Dilek TAPUCU Yücel SAYGIN

## Contents

---

<b>1. INTRODUCTION .....</b>	<b>12</b>
<b>2. PRIVACY- PRESERVING PUBLISHING OF OPINION POLLS .....</b>	<b>14</b>
2.1. Introduction .....	14
2.2. Ensuring Diversity with Multiple Sensitive Attributes .....	15
2.3. Implementation and Experimental Evaluation.....	28
<b>3. PRIVACY-PRECEIVING MULTI-KEYWORD SEARCH .....</b>	<b>19</b>
3.1. Introduction .....	19
3.2. Related Works .....	20
3.3. System and Privacy Requirements Analysis.....	20
3.4. Framework of the Proposed Method .....	28
3.5. The Privacy-Preserving Ranked Multi-Keyword Search .....	22
3.6. Query Randomization.....	28
3.7. Implementation Results .....	28
<b>4. MACHINE LEARNING TECHNIQUES FOR SENTIMENT ANALYSIS.....</b>	<b>27</b>
4.1. Introduction.....	27
4.2. Feature Categorization.....	27
4.3. Sentence Level Analysis for Review Polarity Detection.....	30
4.4. Implementation and Experimental Evaluation.....	30
<b>5. ADAPTATION AND USE OF SUBJECTIVITY LEXICONS FOR DOMAIN DEPENDENT SENTIMENT CLASSIFICATION .....</b>	<b>31</b>
5.1. Introduction .....	32
5.2. Subjectivity Based Feature Extraction .....	32
5.3. Experimental Evaluation .....	35
<b>6. LEARNING DOMAIN SPECIFIC POLARITY LEXICON.....</b>	<b>36</b>
6.1. Introduction .....	38
6.2. Sentiment Analysis with Domain Independent Lexicon.....	38
6.3. Adapting Domain Independent Lexicon.....	40

---

<b>6.4.</b>	<b>Experimental Evaluation .....</b>	<b>41</b>
<b>7.</b>	<b>IMPLEMENTATION AND EVALUATION ON UBIPOL DATA SETS .....</b>	<b>43</b>
7.1	Opinion Mining Engine and its Screenshots.....	46
7.2	Twitter Integration and Opinion Mining on Tweet Data.....	49
7.3	End user Evaluation.....	49
7.3.1	Backend Application Demonstration.....	49
7.3.2	Demonstration to Opinion Mining Engine.....	50
7.3.3	Twitter Extension.....	50
<b>8.</b>	<b>CONCLUSION .....</b>	<b>51</b>

## References

- Orencik C. and Savas E., "Efficient and secure ranked multi-keyword search on encrypted cloud data," in Proceedings of the 2012 Joint EDBT/ICDT Workshops, pp. 186-195, ACM, (2012).
- Gezici, G., Yanikoglu B., Tapucu D., and Saygin Y., "New Features for Sentiment Analysis: Do Sentences Matter?." In SDAD 2012 The 1st International Workshop on Sentiment Discovery from Affective Data, p. 5. (2012).
- Dehkharghani, R., Yanikoğlu, B., Tapucu D., and Saygin Y., "Adaptation and Use of Subjectivity Lexicons for Domain Dependent Sentiment Classification", In SENTIRE 2012 Sentiment Elicitation from Natural Text for Information Retrieval and Extraction, IEEE International Conference on Data Mining (ICDM 2012).
- Demiröz, G., Yanikoğlu B., Tapucu D., and Saygin Y., "Learning domain-specific polarity lexicons." In SENTIRE 2012, Sentiment Elicitation from Natural Text for Information Retrieval and Extraction, IEEE International Conference on Data Mining (ICDM 2012).
- Hussaini M., Kocyigit A., Tapucu D., Yanikoglu B., and Saygin Y., "An aspect-lexicon creation and evaluation tool for sentiment analysis researchers," in ECMLPKDD, 2012.
- Chor B., Kushilevitz E., Goldreich O., and Sudan M., "Private information retrieval," J. ACM, vol. 45, pp. 965-981, November (1998).
- Trostle J. T. and Parrish A., "Efficient computationally private information retrieval from anonymity or trapdoor groups," in ISC'10, pp. 114-128, (2010).
- Vaquero L. M., Rodero-Merino L., Caceres J., and Lindner M., "A break in the clouds: towards a cloud definition," SIGCOMM Comput. Commun. Rev., vol. 39, pp. 50-55, December (2008).
- Chang Y.-C. and Mitzenmacher M., "Privacy Preserving Keyword Searches on Remote Encrypted Data," in Applied Cryptography and Network Security, pp. 442-455, Springer, (2005).
- Liu Q, Wang G, and Wu J., "An efficient privacy preserving keyword search scheme in cloud computing," in Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 02, CSE '09, (Washington, DC, USA), pp. 715-720, IEEE Computer Society, (2009).
- Groth J., Kiayias A., and Lipmaa H., "Multi-query computationally-private information retrieval with constant communication rate," in PKC, pp. 107-123, (2010).
- Freedman M. J., Ishai Y., Pinkas B, and Reingold O., "Keyword search and oblivious pseudorandom functions", in Theory of Cryptography Conference - TCC 2005, pp. 303-324, (2005).
- Boneh D. and Franklin M. K., "Identity based encryption from the weil pairing," IACR Cryptology ePrint Archive, vol. 2001, p. 90, (2001).
- Wang C., Cao N., Li J., Ren K., and Lou W., "Secure ranked keyword search over encrypted cloud data", in ICDCS'10, pp. 253-262, (2010).
- Kuzu M., Islam M. S., and Kantarcioglu M., "Efficient similarity search over encrypted data", in Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12, (Washington, DC, USA), pp. 1156-1167, IEEE Computer Society, (2012).
- Cao N., Wang C., Li M., Ren K., and Lou W., "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in IEEE INFOCOM, (2011).
- Zhang B. and Zhang F., "An efficient public key encryption with conjunctive-subset keywords search," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 262- 267, (2011).
- Wang P., Wang H., and Pieprzyk J., "An efficient scheme of common secure indices for conjunctive keyword-based retrieval on encrypted data," in Information Security Applications, Lecture Notes in Computer Science, pp. 145-159, Springer, (2009).
- Haciguumus H., Iyer B., Li C., and Mehrotra S., "Executing sql over encrypted data in the database-service-provider model", in Proceedings of the 2002 ACM SIGMOD international conference on Management of data, SIGMOD '02, pp. 216-227, ACM, (2002).
- Kinder R, Public Opinions and Political Action, in: The Handbook of Social Psychology, New York, pp. 659-741 (1985).
- Lippmann W., "Public opinion", Transaction Publishers, (1997).
- Gal T.S., Chen Z., Gangopadhyay A., "A Privacy Protection Model for Patient Data with Multiple Sensitive Attributes", in, IGI Global, pp. 28-44, (2008).

- Li Z., Ye X., “Privacy protection on multiple sensitive attributes”, in: Proceedings of the 9th international conference on Information and communications security, Springer-Verlag, Zhengzhou, China, pp. 141-152, (2007).
- L.X.-Y. YANG Xiao-Chun, WANG Bin, YU Ge “Privacy Preserving Approaches for Multiple Sensitive Attributes in Data Publishing, Chinese Journal of Computers, (2008).
- Zhong S., Yang Z., Chen T. “ k-Anonymous data collection”, Information Sciences, 179, 2948-2963 (2009).
- Sweeney L., “Achieving k-anonymity privacy protection using generalization and suppression, Int. J. Uncertain. Fuzziness Knowl.-Based Syst., 10, 571-588, (2002).
- Ninghui L., Tiancheng L., Venkatasubramanian S., “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity”, in: Data Engineering, ICDE 2007. IEEE 23rd International Conference, pp. 106-115, (2007).
- Matatov N., Rokach L., Maimon O., “Privacy-preserving data mining: A feature set partitioning approach”, Information Sciences, 180, pp 2696-2720, (2010).
- Nergiz M.E., Atzori M., Clifton C., “Hiding the presence of individuals from shared databases”, in: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM, Beijing, China, pp. 665-676, (2007).
- Chung K-L., Huang, Y-L., Liu Y-W, Efficient algorithms for coding Hilbert curve of arbitrary-sized image and application to window query, Information Sciences, 177 (2007) 2130-2151.
- LeFevre K., DeWitt D.J., Ramakrishnan R., “Workload-aware anonymization”, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, Philadelphia, PA, USA, pp. 277-286, (2006).
- Ghinita G., Karras P., Kalnis P., Mamoulis N., “Fast data anonymization with low information loss”, in: Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment, Vienna, Austria, pp. 758-769, (2007).
- Aggarwal C.C., “On k-anonymity and the curse of dimensionality”, in: Proceedings of the 31st international conference on Very large databases, VLDB Endowment, Trondheim, Norway, pp. 901-909, (2005).
- [31] Truta T. M., Vinay B., “Privacy Protection: p-Sensitive k-Anonymity Property”, in: Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on, pp. 94-94, (2006).
- Li Z., Ye X., Privacy protection on multiple sensitive attributes, in: Proceedings of the 9th international conference on Information and communications security, Springer-Verlag, Zhengzhou, China, pp. 141-152, (2007).
- Gal T.S., Chen Z., “A. Gangopadhyay, A Privacy Protection Model for Patient Data with Multiple Sensitive Attributes”, in, IGI Global, pp. 28-44, (2008).
- Ahmed, A., Hsinchun, C., Arab, S.: Sentiment analysis in multiple languages: Feature Selection for Opinion classification in Web forums. ACM Transactions on Information Systems 26, 1-34 (2008)
- Bespalov, D., Bai, B., Qi, Y., Shokoufandeh, A.: Sentiment classification based on supervised latent n-gram analysis. In: ACM Conference on Information and Knowledge Management (CIKM) (2011)
- Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines (2001).
- Denecke, K.: How to assess customer opinions beyond language barriers? In: ICDIM. pp. 430-435. IEEE (2008).
- Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06. pp. 417-422 (2006).
- Gindl, S., Weichselbraun, A., Scharl, A.: Cross-domain contextualization of sentiment lexicons. Media (2010).
- Page B.I., Shapiro R.Y., “Effects of Public Opinion on Policy”, The American Political Science Review, 77, pp. 175-190, (1983).
- Grbner, D., Zanker, M., Fliedl, G., Fuchs, M.: Classification of customer reviews based on sentiment analysis. Social Sciences (2012).
- Hatzivassiloglou, V., Mckeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics. pp. 174-181. Association for Computational Linguistics (1997).

- Kim, S.m., Hovy, E., Rey, M.: Automatic detection of opinion bearing words and sentences pp. 61-66.
- Lau, R.Y.K., Lai, C.L., Bruza, P.B., Wong, K.F.: Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 2457{2460. CIKM '11, ACM, New York, NY, USA (2011).
- Mao, Y., Lebanon, G.: Isotonic conditional random elds and local sentiment ow. In: Advances in Neural Information Processing Systems (2007).
- Martineau, J., Finin, T.: Delta t df: An improved feature space for sentiment analysis. In: Adar, E., Hurst, M., Finin, T., Glance, N.S., Nicolov, N., Tseng, B.L. (eds.) ICWSM. The AAAI Press (2009).
- Mcdonald, R., Hannan, K., Neylon, T., Wells, M., Reynar, J.: Structured models for ne-to-coarse sentiment analysis. Computational Linguistics (2007).
- Meena, A., Prabhakar, T.V.: Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. Symposium A Quarterly Journal In Modern Foreign Literatures (2), 573{580 (2007).
- Pang, B., Lee, L.: A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. Cornell University Library (2004).
- Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classication using machine learning techniques. In: Proceedings of EMNLP. pp. 79-86 (2002).
- Taboada, M., Brooke, J., Toloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist. 37(2), 267-307.
- The TripAdvisor website. <http://www.tripadvisor.com> (2011), [TripAdvisor LLC]
- Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: A rating regression approach. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining pp. 783-792 (2010).
- Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005).
- Yu, H.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Proceeding EMNLP 03 Proceedings of the 2003 conference on Empirical methods in natural language processing (2003).
- Zhai, Z., Liu, B., Xu, H., Jia, P.: Grouping product features using semi-supervised learning with soft-constraints. In: Huang, C.R., Jurafsky, D. (eds.) COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China. pp. 1272-1280. Tsinghua University Press (2010).
- Zhang, E., Zhang, Y.: Usc on rec 2006 blog opinion mining. In: TREC (20006).
- Zhao, J., Liu, K., Wang, G.: Adding redundant features for crfs-based sentence sentiment Classi fication. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 117-126 (2008).
- Liu, B., “Sentiment analysis and subjectivity,” Handbook of Natural Language Processing, vol. 2nd ed, 2010.
- Baccianella S., Esuli A., and Sebastiani F., “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” in Proc. of LREC, 2010.
- Liu B., Hu M., and Cheng J., “Opinion observer: analyzing and comparing opinions on the web,” in WWW '05: Proceedings of the 14th international conference on World Wide Web. New York, NY, USA: ACM, 2005, pp. 342–351.
- Ohana B. and Tierney B., “Sentiment classification of reviews using SentiWordNet,” in 9th. IT & T Conference, 2009, p. 13.
- Kao Y. and Lin Z., “A categorized sentiment analysis of chinese reviews by mining dependency in product features and opinions from blogs.” in Web Intelligence, J. X. Huang, I. King, V. V. Raghavan, and S. Rueger, Eds. IEEE, 2010, pp. 456–459.
- Graebner D., Zanker M., Fliedl G., and Fuchsi M., “Classification of customer reviews based on sentiment analysis,” in In 19th Conference on Information and Communication Technologies in Tourism (ENTER). Springer, 2012.



- Taboada M., Brooke J., Tofiloski M., Voll K., and Stede M., “Lexicon- Based Methods for Sentiment Analysis,” *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, Apr. 2011.
- Polanyi L., and Zaenen A., “Contextual Valence Shifters,” in *Computing Attitude and Affect in Text: Theory and Applications*, ser. The Information Retrieval Series, W. B. Croft, J. Shanahan, Y. Qu, and J. Wiebe, Eds. Berlin/Heidelberg: Springer Netherlands, 2006, vol. 20, ch. 1, pp. 1–10.
- Neviarouskaya A., Prendinger H., and Ishizuka M., “Sentifil: Generating a reliable lexicon for sentiment analysis,” in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, IEEE, 2009.
- Qiu G., Bu J., and Chen C., “Expanding domain sentiment lexicon through double propagation.” in *IJCAI*, C. Boutilier, Ed., 2009, pp. 1199–1204.
- Ding X., and Yu P., “A holistic lexicon-based approach to opinion mining.” in *WSDM*, M. Najork, A. Z. Broder, and S. Chakrabarti, Eds. ACM, 2008, pp. 231–240.
- Hamouda A., and Rohaim A., “Reviews classification using sentiwordnet lexicon,” in *Journal on Computer Science and Information Technology (OJCSIT)*, Vol. (2), No.(1), 2011.
- Kaji N. and Kitsuregawa M., “Building lexicon for sentiment analysis from massive collection of HTML documents,” in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 1075–1083.
- Martineau J. and Finin T., “Delta TFIDF: An improved feature space for sentiment analysis,” in *ICWSM*, 2009. [Online]. Available: <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/187/504>
- Pang B. and Lee L., “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proceedings of the ACL*, 2004, pp. 271–278.
- Wang H., Lu Y., and Zhai C., “Latent aspect rating analysis on review text data: A rating regression approach,” *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 783–792, 2010.
- Hatzivassiloglou V. and Mckeown K., “Predicting the semantic orientation of adjectives,” in *Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1997, pp. 174–181.
- Wiebe J. M., “Learning subjective adjectives from corpora,” in *AAAI*, 2000, pp. 735–740.
- Turney P., “Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews,” 2002, pp. 417–424.
- Das S. R., Chen M. Y., Agarwal T. V., Brooks C., Chan Y. shee, Gibson D., D. Leinweber, A. Martinez-jerez, P. Raghuram, S. Rajagopalan, A. Ranade, M. Rubinstein, and P. Tufano, “Yahoo! for amazon: Sentiment extraction from small talk on the web,” in *8th Asia Pacific Finance Association Annual Conference*, 2001.
- Kim S. min, “Determining the sentiment of opinions,” in *Proceedings of COLING*, 2004, pp. 1367–1373.
- Baccianella S., Esuli A., and Sebastiani F., “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” in *Proc. of LREC*, 2010.
- Liu B. and Zhang L., “A survey of opinion mining and sentiment analysis.” in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Springer, 2012, pp. 415–463.
- Fellbaum C., Ed., *WordNet An Electronic Lexical Database*. Cambridge, MA ; London: The MIT Press, May 1998. [Online]. Available: <http://mitpress.mit.edu/catalog/item/default.asp?tttype=2&tid=8106>
- Wilson T., Wiebe J., and Hoffmann P., “Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis,” *Computational Linguistics*, pp. 399–433, 2009.
- Qiu G., Liu B., Bu J., and Chen C., “Expanding domain sentiment lexicon through double propagation,” in *Proceedings of the 21st international joint conference on Artificial intelligence*, ser. *IJCAI’09*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 1199–1204.
- Choi Y. and Cardie C., “Adapting a polarity lexicon using integer linear programming for domain specific sentiment classification,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009, pp. 590–598.

- Dragut E. C., Yu C., Sistla P., and Meng W., “Construction of a sentimental word dictionary,” in Proceedings of the 19th ACM international conference on Information and knowledge management, ser. CIKM '10. New York, NY, USA: ACM, 2010, pp. 1761–1764.
- Lu Y., Castellanos M., Dayal U., and Zhai C., “Automatic construction of a context-aware sentiment lexicon: an optimization approach,” in Proceedings of the 20th international conference on World wide web, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 347–356.
- Martineau J. and Finin T., “Delta TFIDF: An improved feature space for sentiment analysis,” in ICWSM, 2009.
- Popescu A.-M. and Etzioni O., “Extracting product features and opinions from reviews,” in Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 339–346.
- Salton G., Wong A., and Yang C. S., “A vector space model for automatic indexing,” in Communications of the ACM, 1975, pp. 613– 620.
- Paltoglou G., Gobron S., Skowron M., Thelwall M., and Thalmann D., “Sentiment analysis of informal textual communication in cyberspace,” 2010.
- Li S., Lee S. Y. M., Chen Y., Huang C.-R., and Zhou G., “Sentiment classification and polarity shifting,” in Proceedings of the 23rd International Conference on Computational Linguistics, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 635–643.
- Ikeda D., Takamura H., Ratnov L. arie, and Okumura M., “Learning to shift the polarity of words for sentiment classification,” in Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP, 2008).
- Wang H., Lu Y., and Zhai C., “Latent aspect rating analysis on review text data: A rating regression approach,” Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 783–792, 2010.
- Pang B. and Lee L., “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in Proceedings of the ACL, 2004, pp. 271–278.

- Abbreviations and symbols

**Abb.** *[in alphabetic order]*: Full term meaning

**MSA** Multiple Sensitive Attributes

**SBJ.** subjective words

## 1. Introduction

---

This report is the latest version of the D7.3. The main goals of this deliverable are to show our privacy preserving opinion mining solution, to describe opinion mining engine and to explain its algorithms and evaluations. For this engine NLP and machine learning approaches are used. In the first stage, NLP based approach was integrated into the backend of UBIPOL by PDMFC. In this application, domain ontologies were used to explain domain knowledge. For the domain ontologies CORV and SU worked together. During the previous review, reviewers suggested us to integrate twitter into the UbiPOL system. For this purpose, we developed tweet collection and processing tool. Also we implemented supervised machine learning techniques for opinion mining. Twitter was integrated into the opinion mining engine together with policy issues collected through the UbiPOL client.

In **Section 2** we identify the privacy problem regarding public opinions and propose new probabilistic privacy model MSA (Multiple Sensitive Attributes) diversity, specifically defined on datasets with multiple sensitive attributes. We also present a heuristic anonymization technique to enforce MSA-diversity. Experimental results on real data show that our approach clearly outperforms the existing approaches in terms of anonymization accuracy. In **Section 3**, we propose an efficient multi-keyword search scheme that ensures users' privacy against both external adversaries including other authorized users and cloud server itself. The proposed scheme uses cryptographic techniques as well as query and response randomization. Provided that the security and randomization parameters are appropriately chosen, both search terms in queries and returned responses are protected against privacy violations. The scheme implements strict security and privacy requirements that essentially disallow linking queries featuring identical search terms. **Section 4** presents our taxonomy of sentiment analysis features, together with the newly proposed features. **Section 5** describes the adaptation and use of subjectivity lexicons for domain dependent sentiment classification. In **Section 6**, we explain domain specific polarity lexicon with the experimental results and error analysis. **Section 7**, shows implementation and evaluation results on UBIPOL Data sets. In this section, we used end user comment corpuses to show our system's output. First, we showed opinion mining engine then explained tweet extension. Finally, in **Section 8**, we draw some conclusions.

## 2. Privacy-Preserving Publishing of Opinion Polls

### 2.1. Introduction

Governments, political parties, social associations, etc., need to stay in touch with their audiences. Understanding public opinion is essential for a democratic process. Public opinion helps political decision-makers to understand underlying issues that are of utmost importance for them. Issues such as discrimination, gay rights, abortion, cloning, capital punishment, armative action, euthanasia, and national security are examples of hot public opinion topics governments need a comprehensive analysis (Kinder, 1985), (Lippmann,1997). Social research and opinion polls give people the opportunity to express their views regularly on different topics and provide an efficient way to measure public opinion. Since 1973, the European Commission has been monitoring the evolution of public opinion in the Member States, information which helps in the preparation of texts, decision-making and the evaluation of its work. A user profile needs to be constructed for individuals to participate in the public opinion process. These profiles contain valuable data about the user, such as nation, gender, city, and so on. This data may also contain name, address, user's social ID, date of birth and gender. Due to the rapid developments in computer and network technologies, many on-line public opinion polls and mobile-based public opinion systems are used in the opinion process, thus enabling greater participation. Therefore, the public opinion process must guarantee that individuals can express their preferences freely without any threats to their own privacy. Polls done under the risk of identification may not be accurate. For example, Figure 1 shows that in Africa, Asia and the Middle East, attitudes toward homosexuals are generally negative while the European and American voters are generally positive. Voters with Yes/No from an opposing /supporting country may receive public pressure from majority of their countryman if their identities are revealed. If voters are not convinced that such a risk is small, they may not want to reveal their true opinion causing a bias towards the more common attitudes.

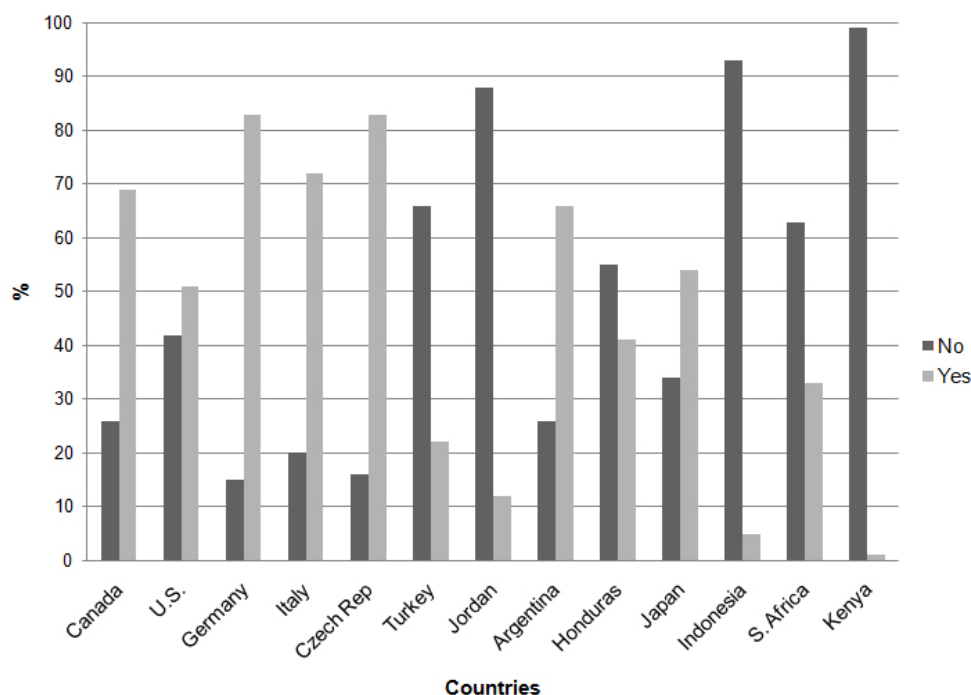


Figure 1: Public opinions on acceptance homosexuality in different countries

In order to publish opinion polls in privacy preserving manner neither the organizing authorities nor any other third party link an opinion to the individual who has cast that opinion assuming some degree of anonymity has to be achieved. As a naive approach, anonymity can be achieved by removing the attributes which uniquely identify individual users such as name, social security number, address, phone number. However, as shown in (Gal et al., 2008), (Li et al., 2007), (Yang, 2008), (Zhong et al., 2009), this approach will not be enough to ensure

anonymity due to the existence of quasi-identifier attributes (QI) which can be used together to identify individuals based on their profile information. Attributes like birth date, gender and ZIP code, when used together, can accurately identify individuals (Sweeney, 2002).

**Table 1: The microdata sample T**

Tuple ID	Quasi-Identifiers (Q-I)		Sensitive Attributes (SA)	
	Age	Zip Code	Issue1 (I1)	Issue2 (I2)
Amy	30	1200	b	w
Bob	20	2400	c	x
Che	23	1500	a	w
Dina	27	3400	c	y

**Table 2: Anonymized data T**

Quasi-Identifiers (Q-I)		Sensitive Attributes (SA)	
Age	Zip Code	Issue1 (I1)	Issue2 (I2)
[20-30]	[1200-3400]	b	w
[20-30]	[1200-3400]	c	x
[20-30]	[1200-3400]	a	w
[20-30]	[1200-3400]	c	y

In this work, we examine a case in which we have a large number of opinions and the data owner needs to publish this data. Adversaries can launch an attack based on user profile and public opinion. We focus on the protection of the relationship between the quasi-identifiers and multiple sensitive attributes. Many works like k-anonymity, l-diversity, t-closeness, etc., have been proposed as a privacy protection model for microdata (Sweeney, 2002), (Ninghui et al., 2007). However, most of models only deal with data with a single sensitive attribute (LeFevre et al., 2002), (Aggarwal, 2005), (Truta et al, 2006), (Nergiz, 2007).

It has been shown in (Gal et al., 2008) that direct application of the techniques proposed for these models creates anonymizations that fail to protect privacy under additional background on non-memberships. As an example, take l-diversity which ensures that each individual can at best be mapped to at least l sensitive values and suppose a data owner has the microdata given in Table 1. Directly applying a single-sensitive attribute l-diversity (SSA-diversity) algorithm on the microdata would result in Table 2 which provides 3-diversity. (E.g., an adversary knowing the public table and seeing Table 2 can at best map, say Amy, to 3 distinct values a, b, and c for issue 1, and to w, x, and y for issue 2.) However, if the adversary also knows that Amy does not vote for c for issue 1, he/she can easily conclude that Amy voted for w for issue 2. Note that public opinion polls collect votes on many issues and it is easy to obtain such non-membership knowledge (compared to membership knowledge) making such attacks a threat in the domain of public opinions.

Our contributions can be summarized as follows:

1. We formally define and introduce MSA-diversity privacy protection model for datasets with multiple sensitive attributes.
2. We design a heuristic anonymization algorithm for MSA-diversity. We borrow ideas from state of the art anonymization techniques such as Hilbert curve anonymization (Ghinita et al., 2007), (Chung et al., 2007) to increase utility.
3. We apply our technique to a real public opinion dataset. We show experimentally that, compared to previously proposed approaches, our model can better bound the probability of disclosure, sometimes at better utility levels.

## **2.2. Ensuring Diversity with Multiple Sensitive Attributes**

In this section, we present a solution to the MSA-diversity problem. To the best of our knowledge, there are two works (Gal, 2011), (Li, 2007) that address privacy issues in publishing private data with multiple sensitive attributes. The first work by (Li, 2007) provides a two-step greedy generalization algorithm. First quasi-identifiers are generalized using a top-down specialization greedy algorithm and second sensitive attributes are masked using a bottom-up local recording algorithm. However, in the public opinion case we have few votes for each issue (sensitive attribute). The masking step on such a case would lead to a huge information loss.

Moreover, the proposed model does not guarantee probabilistic bounds on sensitive information disclosure, thus cannot easily be used for risk/benefit/cost analysis.

**Table 3: Private data T**

Tuple ID	QI		SA	
	Age	Zip Code	Issue1 (I1)	Issue2 (I2)
t1	20	3000	a	w
t2	25	3500	b	x
t3	25	4000	d	w
t4	30	6500	a	y
t5	35	4500	b	y
t6	40	5500	a	y
t7	45	6000	c	z
t8	50	5000	a	x
t9	55	6500	c	w

**Table 4: MSA-diversity published data  $T_2^*$**

Tuple ID	QI		SA	
	Age	Zip Code	Issue1 (I1)	Issue2 (I2)
t1	20-25	3000-4000	a	w
t2	20-25	3000-4000	d	z
t3	20-25	3000-4000	d	x
t6	40-55	5000-6500	a	y
t8	40-55	5000-6500	b	x
t9	40-55	5000-6500	c	w
t4	30-45	4500-6500	a	x
t5	30-45	4500-6500	b	y
t7	30-45	4500-6500	c	z

**Table 5: Gal's et al published data T**

Tuple ID	QI		SA	
	Age	Zip Code	Issue1 (I1)	Issue2 (I2)
t1	20-25	3000-4000	a	w
t2	20-25	3000-4000	d	z
t3	20-25	3000-4000	d	x
t6	40-55	5000-6500	a	y
t7	40-55	5000-6500	c	x
t8	40-55	5000-6500	b	x
t9	40-55	5000-6500	C	w
t4	*	*	*	*
t5	*	*	*	*

The closest to our work is the work by Gal et al (Gal, 2008) which extends the definition of  $l$ -diversity to provide protection against non-memberships attacks. Their model ensures that an individual can at best be linked to at least  $l$  distinct sensitive values and under  $i$  bits of non-membership knowledge, the published data should still satisfy  $l_i$ -diversity. For example in Table 3 and Table 5, each anonymization group satisfies 3-diversity, that is every individual can at best be mapped to at least 3 sensitive values. Even if an adversary knows that, say Linda ( $t_6$ ), does not vote for con issue1, the adversary will still not be sure whether Linda votes for y or x thus the model ensures 2-diversity within the group under one bit of non-membership knowledge. The work also proposes a top-down partitioning algorithm. The algorithm starts with the whole data set as a single group and then splits the group into smaller groups until further splits violates the  $l$ -diversity condition.

We criticize this approach in two ways. First the work does not offer a probabilistic model. That is there is little relation between the privacy parameter  $l$  and the probability of disclosure. For example, the groups in Table 5 is

considered 3-diverse however the probability that Alice ( $t_7$ ) votes for  $c$  on issue1 is  $1/2$ . This makes it difficult to make risk/benefit/cost analysis of publishing private data under a privacy parameter  $\epsilon$  (Nergiz et al., 2007).

Second, partition-based approach has been shown to be too strict resulting in less utilized anonymizations. As an example, consider Tables 3, 4 and 5 . Figure 2 shows a 2D representation of table  $T$ , where the x-dimension is Zip code and y-dimension is Age and tuples are placed according to their Age and zip codes. Partition-based approach works on the private table  $T$  under 3-diversity as follows:

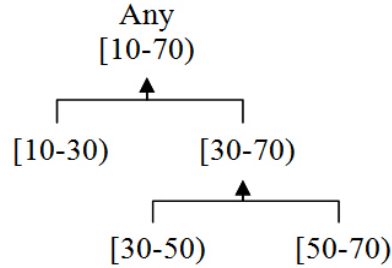


Figure 2: Range-based generalization  $R$

As shown in Figure 3, the algorithm starts by select the attribute with the widest normalized range which in this case is the Age or the Zip code attribute. If the algorithm choose Zip code as a split attribute, it would create two groups  $G_1 = \{t_1, t_2, t_3, t_5, t_8\}$  and  $G_2 = \{t_4, t_6, t_7, t_9\}$ .  $G_2$  does not satisfy 3-diversity. If the algorithm picks attribute Age for the split which gives for 1 the first iteration  $G_3 = \{t_6, t_7, t_8, t_9\}$  and  $G_4 = \{t_1, t_2, t_3\}$  . In 2 the second iteration tuples  $t_4$  and  $t_5$  are excluded. Dashed line represents the partitioning process and the resulting anonymization is  $T_{-1}$  .

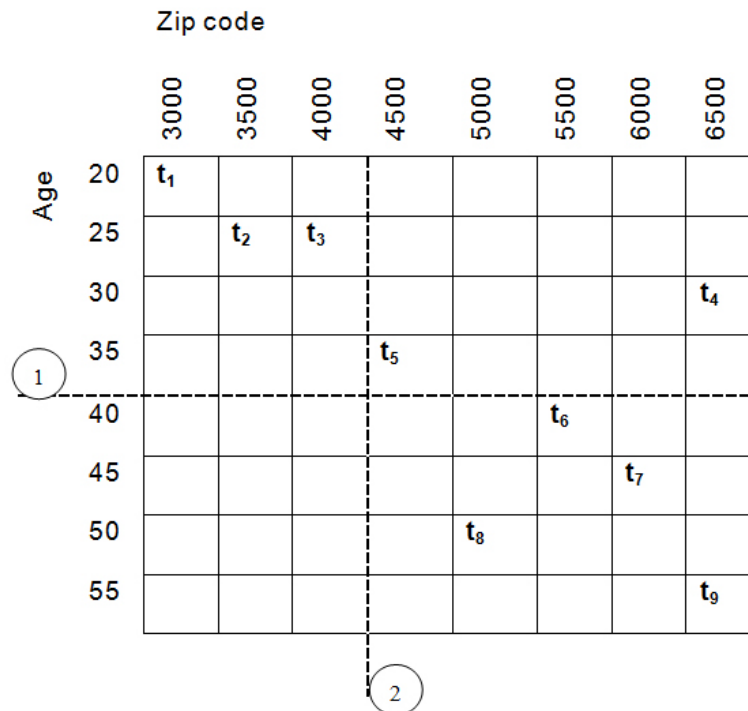


Figure3: 3-diversity groups using Gal's et al model



## 2.3. Implementation and Experimental Evaluation

In this section, we compare our model with state-of-the-art. The Algorithm is implemented in C# and the experiments were run on a Dell 2.4GHz machine with 2GB of memory, running Windows 7. We used MovieLens dataset obtained from the Group-Lens research lab . It contains 10000054 ratings applied to 10681 movies by 71567 users. Ratings for each movie vary from 1 to 7. We used three quasi-identifier attributes Age, Gender and Zip code. We picked seven movies from the dataset that are the most frequently rated among all movies. We marked the movie ratings as sensitive (movies can be thought as issues and actual ratings as opinions). In our experiments, we chose a sub set containing users with ratings for all of the seven movies, therefore our data set become 684 users. We used the discernibility metric (DM), the loss metric (LM) and average query error (AvRE) as information loss metric. We evaluate data accuracy using aggregate query answering as follows. First, we compute the corresponding generalized groups (LeFevre et al., 2006). Second, we process a workload of 684 queries one query for each tuple- on the resulting tables. The effectiveness of generalization is computed by the average relative error.

### 2.3.1. Utility - varying $l$ and $d$

Figure 4 depicts the LM for various  $l$  and  $d$  (number of ratings). Recall that a 0 value of LM means no information loss. Figure 5 reports DM. As can be seen from the figures, an increase in the privacy parameter  $l$  results in more information loss due to the privacy/utility tradeoff. Similarly higher numbers of ratings have similar effect due to curse of dimensionality. Compared to the parameter  $d$ , utility is more sensitive to the changes in  $l$ . For very small and very large  $l$ , the number of ratings has little effect of the utility.

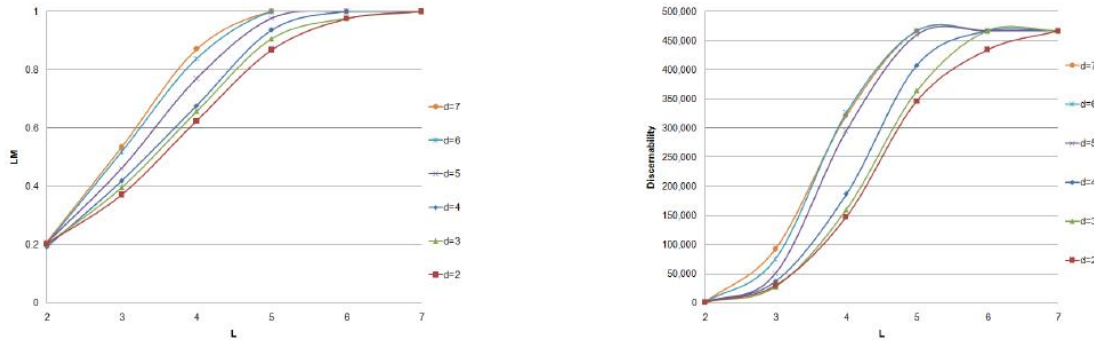


Figure 4:

### 2.3.2. Utility - varying $l$ and $d$

We now compare our approach with the state-of-the-art anonymization algorithm for multiple sensitive attributes by Gal's et al. For Gal's et al model, we assume  $k=1$ . We would like to emphasize that in terms of probability of disclosure, the  $l$ -diversity definition adapted by Gal et al is weaker than  $l$ -mdiversity proposed in Section 2.2. At the same privacy level  $l$  and no non-membership information,  $l$ -diversity by Gal et al ensures the number of distinct sensitive values should be smaller than  $l$ , thus does not guarantee a bound on the probability of disclosure.  $l$ -mdiversity, on the other hand, bounds the probability by  $1/l$ . Moreover, MSA diversity algorithm ensures both privacy metrics. Thus, in our domain, MSA-diversity algorithm offers higher levels of privacy for all  $l$ .

In this paper, we propose a new anonymization model MSA-diversity which limits the probability of disclosure in datasets with multiple sensitive attributes. MSA-diversity is also resistant to attacks by adversaries with nonmembership information. We also present an anonymization algorithm which takes advantage of flexible Hilbert space partitioning to ensure MSA-diversity. We experimentally show that, when compared to state-of-the-art algorithms, the MSA-diversity algorithm offers higher levels of privacy while achieving better utility in most practical privacy settings.

### 3. Privacy-Preserving Multi-Keyword Search

---

#### 3.1. Introduction

Due to the increasing storage and computing requirements of users, everyday more and more data is outsourced to remote, but not necessarily trusted servers. There are several privacy issues regarding to accessing data on such servers; two of them can easily be identified: sensitivity of i) keywords sent in queries and ii) the data retrieved; both need to be hidden. A related protocol, Private Information Retrieval (PIR) (Chor et al., 1988) enables the user to access public or private databases without revealing which data he is extracting. Since privacy is of a great concern, PIR protocols have been extensively studied in the past (Chor et al., 1988), (Trostle, 2010).

Cloud computing has the potential of revolutionizing the computing landscape. Indeed, many organizations that need high storage and computation power tend to outsource their data and services to clouds. Clouds enable its customers to remotely store and access their data by lowering the cost of hardware ownership while providing robust and fast services (Vaquero et al., 2008). Forecasts indicate that by 2014, the cloud services brokerage (CSB) providers, that assist organizations to consume and maintain cloud services, will rise from dozens to hundreds. It is also expected that by 2015, more than half of Global 1000 enterprises will utilize external cloud computing services and by 2016, all Global 2000 will benefit from cloud computing to a certain extent. While its benefits are welcomed in many quarters, some issues remain to be solved before a wide acceptance of cloud computing technology. The security and privacy are among the most important issues (if not the most important). Particularly, the importance and necessity of privacy-preserving search techniques are even more pronounced in the cloud applications. Due to the fact that large companies that operate the public clouds like Google Cloud Platform, Amazon Elastic Compute Cloud or Microsoft Live Mesh may access the sensitive data such as search and access patterns, hiding the query and the retrieved data has great importance in ensuring the privacy and security of those using cloud services.

In this section, we propose an efficient system where any authorized user can perform a search on an encrypted remote database with multiple keywords, without revealing neither the keywords he searches for, nor the information of the documents that match with the query. The only information that the proposed scheme leaks is the access pattern which is leaked by almost all of the practical encrypted search schemes due to efficiency reasons. A typical scenario that benefits from our proposal is that a company outsources its document server to a cloud service provider. Authorized users or customers of the company can perform search operations using certain keywords on the cloud to retrieve the relevant documents. The documents may contain sensitive information about the company, and similarly the keywords a user searches for may give hints about the content of the documents, hence both must be hidden. Furthermore, search terms themselves may reveal sensitive information about the users as well, which is considered to be a privacy violation by users if learned by others.

Our proposed system differs from majority of the previous works which assume that only the data controller queries the database (Chang et al., 2005), (Liu et al., 2009). In contrast to previous works, our proposal facilitates that a group of users can query the database provided that they possess so called trapdoors for search terms that authorize the users to include them in their queries. Another major superiority of the proposed method is the capability of hiding the search pattern which is the equality among the search requests. Moreover, our proposed system is able to perform multiple keyword search in a single query and ranks the results so the user can retrieve only the most relevant matches in an ordered manner.

The contributions of this section can be summarized as follows. Firstly, we introduce formal definitions for the security and privacy requirements of keyword search on encrypted cloud data including hiding the search pattern. We show that linking a query or a response to another leads to correlation attacks that may result in violation of privacy. Secondly, we show how a basic multi-keyword search scheme can be improved to ensure the given privacy and security requirements in the most strict sense. Besides cryptographic primitives, we use query and response randomization to avoid correlation attacks. We provide an extensive analytical study and a multitude of experimental results to support our privacy claims. Thirdly, we propose a ranking method that proves to be efficient to implement and effective in returning documents highly relevant to submitted search terms. Fourthly, we give formal proofs that the proposed scheme is secure in accordance with the defined requirements. Lastly, we implement the proposed scheme and demonstrate that to the best of our knowledge, it is more efficient than existing privacy-preserving multi-keyword search methods in literature.

### 3.2 Related Works

The problem of Private Information Retrieval (PIR), which is a related topic with privacy-preserving keyword search, was first introduced by Chor et al., 1998). In PIR system, user is assumed to know the identifier of the data item he wants to retrieve from the database and receive that data without revealing what he retrieves. Recently (Groth et al. , 2010) propose a multi-query PIR method with constant communication rate. However, the computational complexity of the server in this method is very inefficient to be used in large databases. On the other hand, PIR does not address as to how the user learns which data items are most relevant to his inquiries. For this, an efficient privacy-preserving keyword search scheme is needed.

Efficient privacy-preserving keyword search methods are extensively studied in literature. Traditionally, almost all such schemes have been designed for single keyword search. Ogata and Kurosawa [14] show privacy-preserving keyword search protocol in the random oracle model, based on RSA blind signatures. The scheme requires a public-key operation per item in the database for every query and that must be performed on the user side. (Freedman et al., 2005) propose an alternative implementation for private keyword search that uses homomorphic encryption and oblivious polynomial evaluation methods. The computation and communication costs of this method are quite large since every search term in a query requires several homomorphic encryption operations both on the server and the user side. (Liu et al. , 2009) propose an efficient keyword search scheme utilizing bilinear maps that is based on the public key encryption with keyword search (PEKS) scheme proposed by (Boneh et al., 2006). The scheme is designed for a single user and more importantly, queries in the scheme are generated in a deterministic way, and therefore, cannot hide search pattern. A work proposed by (Wang et al. 2010) allows ranked search over an encrypted database by using inner product similarity. However, this work is also limited only to single keyword search queries. Recently, (Kuzu et al. 2012) propose another single keyword search method that uses locality sensitive hashes (LSH) which reveals search and access patterns but nothing else. Different from the other works, this scheme is a similarity search scheme such that any typo that exists in the query can be handled by the matching algorithm.

A number of privacy-preserving multi-keyword search schemes are also proposed in literature. One of the most related methods to our solution is proposed by (Cao et al. 2011). Similar to our approach presented here, it proposes a method that allows multi-keyword ranked search over encrypted database. In this method, the data controller needs to distribute a symmetric-key which is used in trapdoor generation to all authorized users. Additionally, this work requires keyword elds in the index. This means that the user must know a list of all valid keywords and their positions as a compulsory information to generate a query. This assumption may not be applicable in several cases. Moreover, it is not efficient due to matrix multiplication operations of square matrices where the number of rows is determined essentially by the size of the set of keywords used in searches, which can be in the order of several thousands.

(Zhang and Zhang , 2011) propose a conjunctive keyword search scheme using bilinear pairing based crypto system that does not require keyword fields. Due to the complex bilinear mapping operations and the trapdoor generation operation that must be done on the client side, this scheme is not practical. Moreover, this work is not implemented by the authors therefore, cannot be compared with our proposed work. (Wang et al., 2009) also propose a trapdoorless private multi-keyword search scheme that is proven to be secure under the random oracle model. The scheme uses binary check to test whether the secure index contains the queried keywords, therefore, search is efficient. However there are some security issues that are not addressed in the paper. We adapt their indexing method to our scheme, but we use a different encryption methodology to increase the security and address the security issues that are not considered in that paper.

### 3.3 System and Privacy Requirements

The problem that we consider is privacy-preserving keyword search on private database model, where the documents are simply encrypted with the secret keys unknown to the actual holder of the database (e.g, cloud server). We consider three roles consistent with previous works (Wang et al., 2011):

- Data Controller is the actual entity that is responsible for the establishment of the database. The data controller collects and/or generates the information in the database and lacks the means (or is unwilling) to maintain/operate the database,
- Users are the members in a group who are entitled to access (part of) the information of the database,

- Server is a professional entity (e.g., cloud server) that offers information services to authorized users. It is often required that the server be oblivious to content of the database it maintains, the search terms in queries and documents retrieved.

Given a query from the user, the server searches over the database and returns a list of ordered items. Note that this list does not contain any useful information to the third parties. Upon receiving the list of ordered items, the user selects the most relevant data items and retrieves them.

The privacy definition for search methods in the related literature is that the server should learn nothing but the search results (i.e. access pattern) (Cao et al., 2011). We further tighten the privacy over this general privacy definition and establish a set of privacy requirements for privacy-preserving search protocols. A multi-keyword search method must provide the following user and data privacy properties (First intuitions and then formal definitions are given):

1. (Query Privacy) The query does not leak the information of the corresponding search terms it contains.
2. (Search Term Privacy) Given a valid query for a set of genuine search terms, no one can generate another valid query for a subset of the genuine search terms in the former query.
3. (Search Pattern Privacy) Equality between two search requests cannot be verified by analyzing the queries or the returned list of ordered matching results.
4. (Non-Impersonation) No one can impersonate a legitimate user.

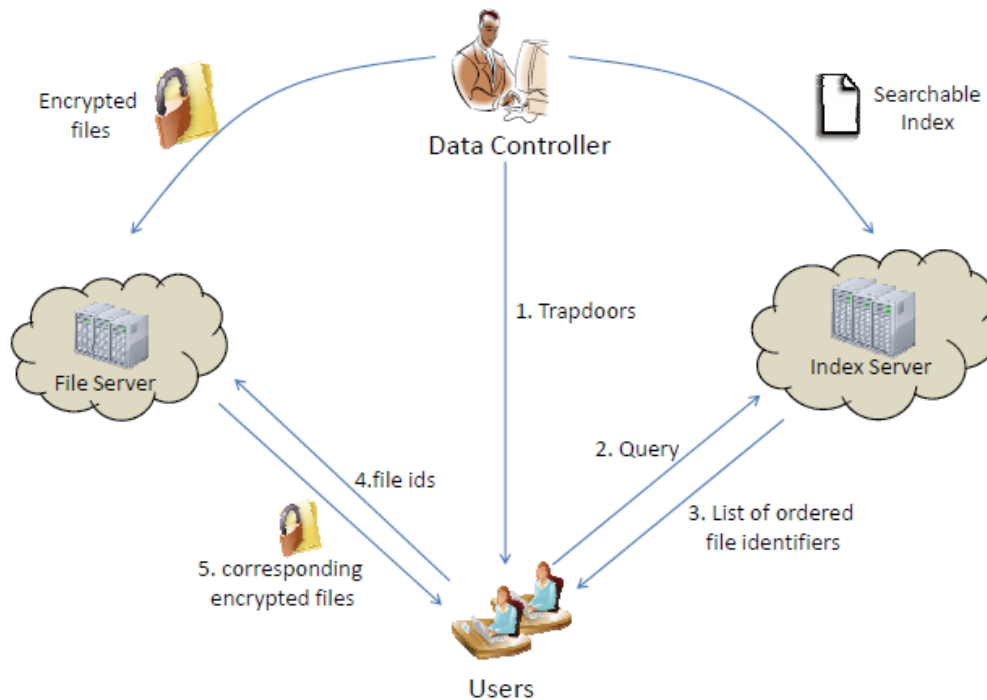


Figure5: Architecture of the search method

### 3.4 Framework of the Proposed Method

The previous section introduces the three roles that we consider: Data Controller, Users and Server. Due to the privacy concerns that is explained, we utilize two servers namely: index server and file server. The overview of the proposed system is illustrated in 1. We assume that the parties are semi-honest ("honest but curious") and do not collude with each other to bypass the security measures, two assumptions which are consistent with most of the previous works.

In Figure 5, steps and typical interactions between the participants of the system are illustrated. In an offline stage, the data controller creates a search index element for each document. The search index file is created using a secret key based trapdoor generation function where the secret keys<sub>1</sub> are only known by the data controller.

Then, the data controller uploads this search index file to the index server and the encrypted documents to the file server. We use symmetric-key encryption as the encryption method since it can handle large document sizes efficiently. This process is referred as the index generation henceforth and the trapdoor generation is considered as one of its steps.

When a user wants to perform a keyword search, he first connects to the data controller. He learns the trapdoors (cf. Step 1 in Figure 5) for the keywords he wants to search for, without revealing the keyword information to the data controller. Since the user can use the same trapdoor for many queries containing the corresponding search term, this operation does not need to be performed every time the user performs a query.

Alternatively, the user can request all the trapdoors in advance and never connects again to the data controller for trapdoors. This can differ according to the application and the users' requirements. After learning the trapdoor information, the user generates the query (referred as query generation henceforth) and submits it to the index server (cf. step 2 in Figure 5). In return, he receives metadata<sub>2</sub> for the matched documents in a rank ordered manner as will be explained in subsequent sections. Then the user retrieves the encrypted documents from the file server after analyzing the metadata that basically conveys a relevancy level of the each matched document, where the number of documents returned is specified by the user.

### 3.5 The Privacy-Preserving Ranked Multi-Keyword Search

In this section, we provide the **details** for the crucial steps in our proposal, namely index generation, trapdoor generation, and query generation.

- **Index Generation (basic scheme)**

Recently Wang et al., 2009 proposed a conjunctive keyword search scheme that allows multiple-keyword search in a single query. We use this scheme as the base of our index construction scheme. The original scheme uses forward indexing, which means that a searchable index file element for each document is maintained to indicate the search terms existing in the document. In the scheme of Wang et al. 2009, a secret cryptographic hash function, that is shared between all authorized users, is used to generate the searchable index. Using a single hash function shared by several users forms a security risk since it can easily leak to the server. Once the server learns the hash function, he can break the system if the input set is small. The following example illustrates a simple attack against queries with few search terms.

Example 1 There are approximately 25000 commonly used words in English and users usually search for a single or two keywords. For such small input sets, given the hashed trapdoor for a query, it will be easy for the server to identify the queried keywords by performing a brute-force attack. For instance, assuming that there are approximately 25000 possible keywords in a database and a query submitted by a user involves two keywords, there will be  $25000^2 < 2^{28}$  possible keyword pairs. Therefore, approximately  $2^{27}$  trials will be sufficient to break the system and learn the queried keywords.

**Table 6: Index Genetation Algorithm**

---

**Algorithm 1** Index Generation

---

Require:  $\mathcal{R}$  : the document collection,  $K_{id}$ : secret key for the bin with label  $id$

```

for all documents  $R_i \in \mathcal{R}$  do
  for all keywords  $w_{ij} \in R_i$  do
     $id \leftarrow \text{GetBin}(w_{ij})$ 
     $x_{ij} \leftarrow \text{HMAC}_{K_{id}}(w_{ij})$ 
     $I_{ij} \leftarrow \text{Reduce}(x_{ij})$ 
  end for
  index entry  $\mathcal{I}_{R_i} \leftarrow \odot_j I_{ij}$ 
end for
return  $\mathcal{I} = \{\mathcal{I}_{R_1}, \dots, \mathcal{I}_{R_\sigma}\}$ 

```

---

- **Query Generation**

The search index file of the database is generated by the data controller using secret keys. A user who wants to include a search term in his query, needs the corresponding trapdoor from the data controller since he does not know the secret keys used in the index generation. Asking for the trapdoor openly would violate the privacy of the user against the data controller, therefore a technique is needed to hide the trapdoor asked by the user from the data controller.

Bucketization is a well-known data partitioning technique that is frequently used in literature (Haciguumus et al., 2002). We adopt this idea to distribute keywords into a fixed number of bins depending on their hash values. More precisely, every keyword is hashed by a public hash function, and certain number of bits in the hash value is used to map the keywords to these bins. The number of bins and the number of keywords in each bin can be adjusted according to security and efficiency requirements of the system.

In our proposal for obtaining trapdoors, we utilize a public hash function with uniform distribution. All keywords that exist in a document are mapped by the data controller to one of those bins using the GetBin function. The query generation method which is summarized in Algorithm 2, works as follows. When the user connects to the data controller to obtain the trapdoor for a keyword, he first calculates the bin IDs of keywords and sends these values to the data controller. The data controller then returns the secret keys of the bins requested for, which can be used by the user to generate the trapdoors for all keywords in these bins. Alternatively, the data controller can send trapdoors of all keywords in corresponding bins resulting in an increase in the communication overhead. However, the latter method relieves the user of computing the trapdoors. After obtaining the trapdoors, the user can calculate the query in a similar manner to the method used by the data controller to compute the search index.

**Table 7: Query Generation Algorithm**

---

**Algorithm 2** Query Generation

---

Require: a set of search terms  $\{w'_1, \dots, w'_n\}$   
 for all search terms  $w'_i$  do  
    $id \leftarrow \text{GetBin}(w'_i)$   
   if  $K_{id} \notin$  previously received keys then  
   send  $id$  to Data Controller  
   get  $K_{id}$  from Data Controller  
   end if  
    $x_i \leftarrow \text{HMAC}_{K_{id}}(w'_i)$   
    $I_i \leftarrow \text{Reduce}(x_i)$   
 end for  
 query  $Q \leftarrow \odot_i I_i$   
 return  $Q$

---

- **Oblivious Search on the Database**

A user's query, in fact, is just an  $r$ -bit binary sequence (independent of the number of search terms in it) and therefore, searching consists of as simple operations as binary comparison only. If the search index entry of the document ( $\mathcal{I}_R$ ) has 0 for all the bits, for which the query ( $Q$ ) has also 0, then the query matches to that document as shown in Equation. Note that given a query, it should be compared with search index entry of each document in the database.

$$\text{result}(Q, \mathcal{I}_R) = \begin{cases} \text{match} & \text{if } \forall j \ Q^j = 0 \Rightarrow \mathcal{I}_R^j = 0 \\ \text{not match} & \text{otherwise} \end{cases}$$

- **Document Retrieval**

The index server returns the list of pseudo identifiers of the matching documents. If a single server is used for both search and file retrieval, it can be possible to correlate the pseudo identifiers of the matching documents and the identifiers of the actual encrypted files retrieved. Furthermore, this may also leak the search pattern that we want to hide. Therefore, we use a two-server system similar to the one proposed in [18] where the two servers are both semi-honest and do not collaborate. This method leaks the access pattern only to the file server and not

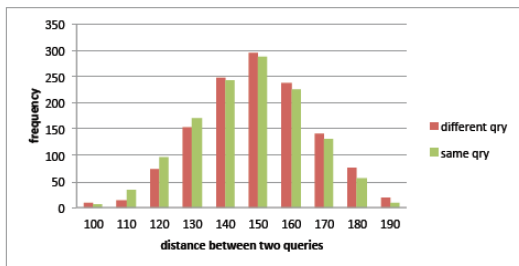
to the index server, hence prevent any possible correlation between search results and encrypted documents retrieved. Subsequent to the analyzes of the metadata retrieved from the index server, the user requests a set of encrypted files from the file server. The file server returns the requested encrypted files. Finally user decrypts the files and learns the actual documents. Key distribution of the document decryption keys can be done using state of the art key distribution methods and is not within the scope of this work.

In case access pattern needs also to be hidden, traditional PIR methods [2-6] or Oblivious RAM [28] can be utilized for the document retrieval process instead. However these methods are not practical even for medium sized datasets due to incurred polylogarithmic overhead.

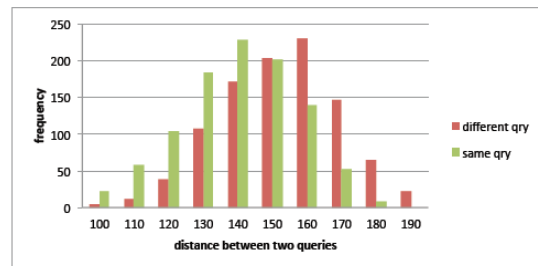
### 3.6 Query Randomization

Search pattern is the information of equality among the keywords of two queries that can be inferred by linking one query to another. If an adversary can test the equality of two queries, he may learn the most frequent queries and correlate with frequently searched real keywords that may be learned from statistics such as Google Trends [29]. The proposed basic scheme fails to hide the search pattern since the search index entries are generated in a deterministic way. Any two queries created from the identical keywords will be exactly the same. In order to hide the search pattern of a user, we introduce randomness into the query generation phase. This process is known as query randomization, which should be carefully implemented so that the queries do not leak information about the search patterns. In this section, we analytically demonstrate the effectiveness of the proposed query randomization method. Note that the query randomization does not change the response to a given query.

To demonstrate the usefulness of our analysis, we conducted an experiment using synthetic query data for the case, where adversary does not know the number of genuine search terms in a query. We generate a synthetic data for a set of queries with the parameters  $V = 30$  and  $U = 60$  being fixed. The set contains a total of 250 queries, where the first 50 queries contain 2 genuine search terms each, the second 50 queries contain 3 genuine search terms each, and so on. And finally, the last set of 50 queries contains 6 genuine search terms each. We create another set which contains only 5 queries, which include 2, 3, 4, 5 and 6 genuine search terms, respectively. The distances between pairs of queries, in which one query is chosen from the former set and the second one from the latter, are measured to obtain a histogram as shown in Figure 6(a). Consequently, a total of  $250 \cdot 5 = 1250$  distances are measured. We also obtain another histogram featuring a total of 1250 distances between pairs of queries, whereby queries in a pair contain the same genuine search terms with different dummy keywords. Both histograms are given in Figure 6(a), where it can be observed that adversary cannot do better than a random guess to identify whether given two queries contain the same genuine search terms or not. We also conducted a similar experiment, in which we assume that the adversary has the knowledge of the number of search terms in a query. We generate a set containing a total of 1000 queries, whose subsets with 200 queries each contain 2, 3, 4, 5 and 6 genuine search terms, respectively. We then create a single query with 5 genuine search terms. We measured the distances of the single query to all 1000 queries in the former set of queries to create a histogram (i.e., a total of  $200 \cdot 5 = 1000$  distances are measured). We compared this with the histogram for 1000 measurements of the distance between two queries with five identical search terms as shown in Figure 6(b). As can be observed from the histogram in Figure 6(b), 20% of the time, distances between two queries are 150 and they are totally indistinguishable.



(a) Histogram for the distances for two arbitrary queries and for two queries that are generated from the same search terms



(b) Histogram for the distances for two arbitrary queries and for two queries that are generated from the same search terms where the number of search terms in the query is 5

Figure 6: Histograms for the Hamming distances between queries



The privacy-preserving multi-keyword search (MKS) method must provide the user and data privacy requirements specified by definitions in Section 3. This section is devoted to the proofs that the proposed method indeed satisfies these privacy requirements. In proofs, we assume that the randomization parameters are selected appropriately by taking into consideration of the database or search statistics.

The proposed method is semantically secure against chosen keyword attacks under indistinguishability of ciphertext from ciphertext (ICC). The formal proof is provided in Theorem 1 which we adopt their indexing method; therefore, we omit this proof here. Intuitively, the proof is based on the property that since the HMAC function is a random function, hash values of any two different keywords will be two random numbers and be independent of each other. Therefore, given two keyword lists  $L_0$ ,  $L_1$  and an index  $lb$  for the keyword list  $L_b$  where  $b \in \{0, 1\}$ , it is not possible for an adversary to guess the value  $b$  with probability greater than  $1/2$ . The security against chosen keyword attacks is required, but not sufficient for the privacy-preserving search.

### 3.7 Implementation Results

The entire system is implemented by Java language using socket programming on an iMac with Intel Core i7 processor of 2.93 GHz. Considering analyzing a document for finding the keywords in it, is out of the scope of this work, a synthetic database is created by assigning random keywords with random term frequencies for each document. The HMAC function produces outputs, whose size ( $l$ ) is 336 bytes (2688 bit), which is generated by concatenating different SHA2-based HMAC functions. We choose  $d = 6$  so that after the reduction phase the result is reduced to one-sixth of the original result; therefore the size of each database index entry and ( $r$ ) is 56 bytes (448 bits).

In our experiments, we used different datasets with different number of documents (from 2000 to 10000 documents). The timing results for creating the queries are obtained for documents with 30 genuine search terms and 60 random keywords each using ranking technique with different rank levels for parameters  $q = 1$  and  $f = 5$  in Figure 7(a). Considering that index generation is performed only occasionally (if not once) by the data controller and that index generation problem is of highly parallelized nature, the proposed technique presents a highly efficient and practical solution to the described problem.

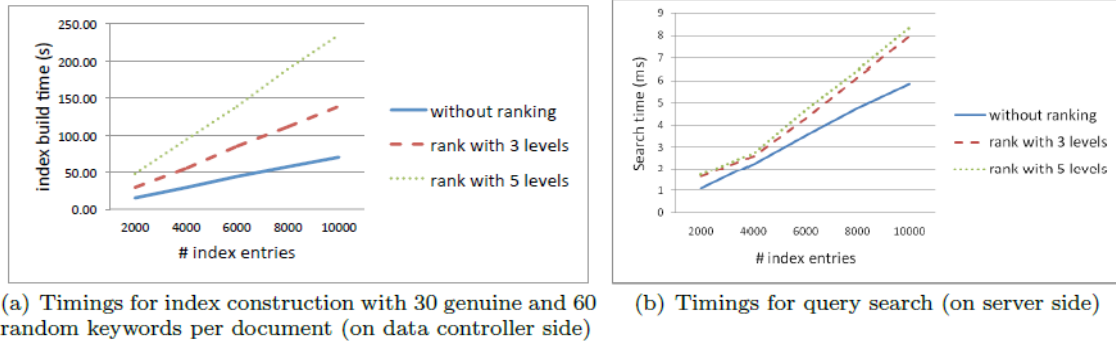


Figure 7: Timing results

Figure 7(b) demonstrates the server timings for a search with different rank levels. As can be observed from the graphic in Figure 7(b), time spent by the server per query is quite low, rendering high-throughput processing of user queries possible. By parallelization and native language support, the throughput can be increased several orders of magnitude.

Most of the privacy-preserving search methods that exist in literature are only capable of single keyword search. The problem that we consider is multi-keyword search; therefore, we did not provide a comparison with the works that consider single keyword search. A very recent work by Cao et al, 2011, is the closest work to our proposed method. Our implementations show that our method is one to two orders of magnitude faster than the method in (Cao et al., 2011) in both offline and online operations. The index construction method of (Cao et al., 2011), takes about 4500 s for 6000 documents where we need 140 s in the highest rank level. Similarly that work requires 600 ms to perform a search over 6000 documents where we need only 4.7 ms.. The tests in [19] were done on an equivalent computer, Intel Xeon processor 2.93 GHz. Among the other existing multi-keyword solutions, bilinear pairing based methods such as Wang et al, 2009 provide only theoretical solutions. This method in Zang et al., 2011 is not implemented due to excessive computational requirements hence, cannot be



compared with our proposed work. The work of Wang et. al., 2009) which is the inspiration for our proposed method, provides a faster solution than our work since they do not use additional fake entries or dummy keywords. However, that work does not satisfy some of the privacy requirements such as hiding search pattern privacy. The low time requirements on the data controller side enable processing multiple requests with high-throughput. Note that the programs used in the experiments are developed in Java language for portability reasons and unoptimized. Further optimization or support of native code or parallel implementation will further increase the performance of the proposed system.

## 4. Machine Learning Features for Sentiment Analysis

### 4.1. Introduction

We performed a preliminary evaluation using the TripAdvisor dataset to see the effect of sentence level features on polarity classification. Throughout the evaluation, we observed a small improvement in classification accuracy due to the newly proposed features. Our initial results showed that the sentences do matter and they need to be explored in larger and more diverse datasets such as blogs. Moreover, the benefit of these features is not limited to improving sentiment classification accuracy. In fact, sentence level features can be used to identify the essential sentences in the review which could further be used in review summarization.

### 4.2. Feature Categorization

We define an extensive set of 19 features that can be grouped in four categories:

- (1) basic features,
- (2) features based on subjective sentence occurrence statistics,
- (3) delta-tf-idf weighting of word polarities, and
- (4) sentence-level features.

These features are listed in Table 7 and using the notations given below and some basic definitions provided in Table 8, they are defined formally in Tables.

**Table 7. Summary Feature Descriptions for a Review R**

Group Name	Feature	Name
<b>Basic</b>	<b>F1</b>	Average review polarity
	<b>F2</b>	Review purity
<b>Occurrence of subj. words</b> <b>Subjective Words</b>	<b>F3</b>	Freq. of subjective words
	<b>F4</b>	Avg. polarity of subj. words
	<b>F5</b>	Std. of polarities of subj. words
<b><math>\Delta TF\_IDF</math></b>	<b>F6</b>	Weighted avg. polarity of subj. words
	<b>F7</b>	Scores of subj. words
<b>Punctuation</b>	<b>F8</b>	# of Exclamation marks
	<b>F9</b>	# of Question marks
<b>Sentence Level</b>	<b>F10</b>	Avg. First Line Polarity
	<b>F11</b>	Avg. Last Line Polarity
	<b>F12</b>	First Line Purity
	<b>F13</b>	Last Line Purity
	<b>F14</b>	Avg. pol. of subj. sentences
	<b>F15</b>	Avg. pol. of pure sentences
	<b>F16</b>	Avg. pol. of non-irrealis sentences
	<b>F17</b>	$\Delta TF\_IDF$ weighted polarity of first line
	<b>F18</b>	$\Delta TF\_IDF$ scores of subj. words in the first line
	<b>F19</b>	Number of sentences in review

A review  $R$  is a sequence of sentences  $R = \{S_1, S_2, S_3, \dots, S_M\}$  where  $M$  is the number of sentences in  $R$ . Each sentence  $S_i$  in turn is a sequence of words, such that  $S_i = w_{i1}, w_{i2}, \dots, w_{iN(i)}$  where  $N(i)$  is the number of words in  $S_i$ . The review  $R$  can also be viewed as a sequence of words  $w_1..w_T$ , where  $T$  is the total number of words in the review.

In Table 8, subjective words (SBJ) are defined as all the words in SentiWord-Net that has a dominant negative or positive polarity. A word has dominant positive and negative polarity if the sum of its positive and negative polarity values is greater than 0.5 (Zhang et al., 2006). SubjW( $R$ ) is defined as the most frequent subjective words in SBJ (at most 20 of them) that appear in review  $R$ . For a sentence  $S_i \in R$ , the average sentence polarity is used to determine subjectivity of that sentence. If it is above a threshold, we consider the sentence as subjective,

forming subj E (R). Similarly, a sentence  $S_i$  is pure if its purity is greater than a fixed threshold  $\beta$ . We experimented with different values of  $\beta$  and for evaluation we used  $\beta = 0.8$ . These two sets form the subS(R) and pure(R) sets respectively.

We also looked at the effect of first and last sentences in the review, as well as sentences containing irrealis words. In order to determine irrealis sentences, the existence of the modal verbs 'would', 'could', or 'should' is checked. If one of these modal verbs appear in the sentence then these sentences are labeled as irrealis similar to (Taboada et al., 2008).

**Table 8. Summary Feature Descriptions for a Review R**

M	the total number of sentences in R
R	the total number of words in R
T	set of known subjective words
SBJ W(R)	set of most frequent subjective words from SBJ, in R (max 20)
subjS(R)	set of subjective sentences in R
pure(R)	set of pure sentences in R
nonIr(R)	set of non-irrealis sentences in R

## Basic Features

For our baseline system, we use the average word polarity and purity defined in f. As mentioned before, these features are commonly used in word polarity based sentiment analysis. In our formulation  $pol(w_j)$  denotes the dominant polarity of  $w_j$  of R, as obtained from SentiWordNet, and  $|pol(w_j)|$  denotes the absolute polarity of  $w_j$ .

**Table 9. Basic Features for a review R**

F1	Average review polarity	
F2	Review purity	$\frac{\sum_{j=1..T} pol(w_j)}{\sum_{j=1..T}  pol(w_j) }$

## Frequent Subjective Words

The features in this group are derived through the analysis of subjective words that frequently occur in the review. For instance, the average polarity of the most frequent subjective words (feature F4) aims to capture the frequent sentiment in the review, without the noise coming from all subjective words. The features were defined before in some previous work (Denecke et al., 2008); however, to the best of our knowledge, they considered all words, not specifically subjective words.

**Table 10. Features Related to Frequency and Subjectivity**

F3	Freq. of subjective words	$\frac{ SubjW(R) }{ R }$
F4	Avg. polarity of subj. words	$\frac{1}{ SubjW(R) } \sum_{w_j \in SubjW(R)} pol(w_j)$
F5	Stdev. of polarities of subj. words	$\sqrt{\frac{1}{ SubjW(R) } \sum_{w_j \in SubjW(R)} (pol(w_j) - F4)^2}$

### $\Delta tf * idf$ Features

We compute the  $\Delta tf * idf$  scores of the words in SentiWordNet from a training corpus in the given domain, in order to capture domain specificity (Martineau et al., 2009). For a word  $w_i$ ,  $\Delta tf * idf(w_i)$  is defined as:

$$\Delta tf * idf(w_i) = tf * idf(w_i, +) - tf * idf(w_i, -)$$

If it is positive, it indicates that a word is more associated with the positive class and vice versa, if negative. We computed these scores on the training set which is balanced in the number of positive and negative reviews.

Then, we sum up the  $\Delta tf * idf$  scores of these words (feature F6). By doing this, our goal is to capture the difference in distribution of these words, among positive and negative reviews. The aim is to obtain context-dependent scores that may replace the polarities coming from SentiWordNet which is a context-independent lexicon (Esulu et al., 2006). With the help of context-dependent information provided by  $\Delta tf * idf$  related features, we expect to better differentiate the positive reviews from negative ones. We also tried another feature by combining the two information, where we weighted the polarities of all words in the review by their  $\Delta tf * idf$  scores (feature F7).

**Table 11.  $\Delta tf * idf$  Features**

F6	$\Delta tf * idf$ scores of all words	
F7	Weight. avg. pol. of all words	

### Punctuation Features

We have two features related to punctuation. These two features were suggested in (Denecke et al., 2008) and since we have seen that they could be useful for some cases we included them in our sentiment classification system.

**Table 12. Punctuation Features**

F8	Number of exclamation marks in the review
F9	Number of question marks in the review

### Sentence Level Features

Sentence level features are extracted from some specific types of sentences that are identified through a sentence level analysis of the corpus. For instance the first and last lines polarity/purity are features that depend on sentence position; while average polarity of words in subjective/pure etc. sentences are new features that consider only subjective or pure sentences respectively.

**Table 13.  $\Delta tf * idf$  Features**

F10	Avg. First Line Polarity	
F11	Avg. Last Line Polarity	
F12	First Line Purity	$\frac{\sum_{j=1 \dots N(1)} pol(w_{j1})}{\sum_{j=1 \dots N(1)}  pol(w_{j1}) }$
F13	Last Line Purity	$\frac{\sum_{j=1 \dots N(M)} pol(w_{jM})}{\sum_{j=1 \dots N(M)}  pol(w_{jM}) }$
F14	Avg. pol. of subj. sentences	
F15	Avg. pol. of pure sentences	
F16	Avg. pol. of non-irrealis sentences	
F17	$\Delta tf * idf$ weighted polarity of 1st line	$\sum_{j=1 \dots T} \Delta tf * idf(w_{1j}) * pol(w_{1j})$
F18	$\Delta tf * idf$ Scores of 1st line	$\sum_{j=1 \dots T} \Delta tf * idf(w_{1j})$
F19	Number of sentences in review	$M$

### 4.3. Sentence Level Analysis for Review Polarity Detection

We tried three different approaches in obtaining the review polarity. In the first approach, each review is pruned to keep only the sentences that are possibly more useful for sentiment analysis. For pruning, thresholds were set separately for each sentence level feature. Sentences with length of at most 12 words are accepted as short and sentences with absolute purity of at least 0.8 are defined as pure sentences. For subjectivity of the sentences, we adopted the same idea that was mentioned in (Zhang et al., 2006) and applied it on not words, but sentences in this case.

Pruning sentences in this way resulted in lower accuracy in general, due to loss of information. Thus, in the second approach, the polarities in special sentences (pure, subjective, short or no irrealis) were given higher weights while computing the average word polarity. In effect, other sentences were given lower weight, rather than the more severe pruning. In the final approach that gave the best results, we used the information extracted from sentence level analysis as features used for training our system. We believe that our main contribution is

the introduction and evaluation of sentence-level features; yet other than these, some well-known and commonly used features are integrated to our system, as explained in the next section.

Our approach depends on the existence of a sentiment lexicon that provide information about the semantic orientation of single or multiple terms. Specifically, we use the SentiWordNet where for each term at a specific function, its positive, negative or neutral appraisal strength is indicated (e.g. "good,ADJ, 0.5)

#### **4.4. Implementation and Experimental Evaluation**

In this section, we provide an evaluation of the sentiment analysis features based on word polarities. We use the dominant polarity for each word (the largest polarity among negative, objective or positive categories) obtained from sentiWord- Net. We evaluate the newly proposed features and compare their performance to a baseline system. Our baseline system uses two basic features which are the average polarity and purity of the review. These features are previously suggested in (Ahmed et al., 2008) and (Zhai et al., 2010) widely used in word polarity-based sentiment analysis. They are defined in Table 3 for completeness. The evaluation procedure we used in our experiments is described in the following subsections.

##### **Dataset**

We evaluated the performance of our system on a sentimental dataset, TripAdvisor that was introduced by (<http://www.tripadvisor.com>) and, (Wang et al., 2010) respectively. The TripAdvisor corpus consists of around 250.000 customer-supplied reviews of 1850 hotels. Each review is associated with a hotel and a star-rating, 1-star (most negative) to 5-star (most positive), chosen by the customer to indicate his evaluation. We evaluated the performance of our approach on a randomly chosen dataset from TripAdvisor corpus. Our dataset consists of 3000 positive and 3000 negative reviews. After we have chosen 6000 reviews randomly, these reviews were shuffled and split into three groups as train, validation and test sets. Each of these datasets have 1000 positive and 1000 negative reviews.

We computed our features and gave labels to our instances (reviews) according to the customer-given ratings of reviews. If the rating of a review is bigger than 2 then it is labeled as positive, and otherwise as negative. These intermediate files were generated with a Java code on Eclipse and given to WEKA (Witten et al., 2005) for binary classification.

##### **Sentiment Classification**

Initially, we tried several classifiers that are known to work well for classification purposes. Then, according to their performances we decided to use Support Vector Machines (SVM) and Logistic regression. SVMs are known for being able to handle large feature spaces while simultaneously limiting overfitting, while Logistic Regression is a simple, and commonly used, well-performing classifier. The SVM is trained using a radial basis function kernel as provided by Lib-SVM (Chang et al., 2001). For LibSVM, RBF kernel worked better in comparison to other kernels on our dataset. Afterwards, we performed grid-search on validation dataset for parameter optimization.

##### **Experimental Results**

In order to evaluate our sentiment classification system, we used binary classification with two classifiers, namely SVMs and Logistic Regression. The reviews with star rating bigger than 2 are positive reviews and the rest are negative reviews in our case, since we focused on binary classification of reviews. Apart from this, we also looked at the importance of the features. The importance of the features will be stated with the feature ranking property of WEKA as well as the gradual accuracy increase, as we add a new feature to the existing subset of features.

For these results, we used grid search on validation set. Then, by these optimum parameters, we trained our system on training set and tested it on testing set.

**Table 14. The Effects of Feature Subsets on TripAdvisor Dataset**

Feature Subset	Accuracy (SVM)	Accuracy (Logistic)
Basic (F1,F2)	79.20%	79.35%
Basic (F1,F2) + _TF _ IDF (F6,F7)	80.50%	80.30%
Basic (F1,F2) + _TF _ IDF (F6,F7) + Freq. of Subj. Words (F3)	80.80%	80.05%
Basic (F1,F2) + _TF _ IDF (F6,F7) + Freq. of Subj. Words (F3) + Punctuation (F8,F9)	80.20%	79.90%
Basic (F1,F2) + _TF _ IDF (F6,F7) + ... Occur. of Subj. Words (F3-F5)	80.15%	79.00%
All Features (F1-F19)	80.85%	81.45%

**Table 15. Comparative Performance of Sentiment Classification System on TripAdvisor Dataset**

Previous Work	Dataset	F-measure	Error Rate
Gindl et al (2010)	1800	0.79	-
Bespalov et al (2011)	96000	-	7.37
Peter et al (2011)	103000	0.82	-
Grabner et al (2012)	1000	0.61	-
<b>Our System (2012)</b>	<b>6000</b>	<b>0.81</b>	<b>-</b>

## 5. Adaptation and Use of Subjectivity Lexicons for Domain Dependent Sentiment Classification

---

### 5.1 Introduction

We study the effect of subjectivity-based features on sentiment classification on two lexicons and also propose new subjectivity-based features for sentiment classification. The subjectivity-based features we experiment with are based on the average word polarity and the new features that we propose are based on the occurrence of subjective words in review texts.

In this section, we address a sub-problem of sentiment analysis, namely classifying reviews as either positive or negative. For the classification of reviews, we use subjectivity-based features. Our contributions could be summarized as:

- (1) Proposing new subjectivity-based features for sentiment classification
- (2) Combining domain independent and domain specific subjectivity-based features and evaluating them.

The first set of the subjectivity based features that we evaluated is based on word polarities obtained from a domain independent lexicon and the second one is based on seed word sets adapted to a specific domain. We used the well-known polarity lexicon SentiWordNet, built by Baccianella et. al. (Baccianella et al., 2010), for extracting domain-independent subjectivity features.

In addition, a domain-specific lexicon of subjective words is used for extracting domain-specific features. To build this lexicon, we selected a subset of the words automatically from the domain-independent subjectivity lexicon built by Hu and Liu (Hu et al., 2005) that is referred as Initial SeedWords in this work, for two different domains; namely the hotel and movie domains. We refer to this domain-specific lexicon of subjective words as SubjWords throughout the section. The domain dependent features are based on the occurrence frequencies of SubjWords in the reviews. The proposed approach gives a small improvement over the baseline and achieves results compared to other findings in the literature such as (Ohana et al., 2009).

Our approach to sentiment analysis is a lexicon-based approach, therefore, we mostly report similar works in this section. Lin Pan (Lin et al., 2010) worked on Chinese language reviews using two sets of positive and negative words, each of which includes more than 4000 words. This work was featurebased and used some predefined templates in sentences. It was applied on different review categories such as hotel review and was able to achieve accuracies higher than 85% in some cases. Graebner et.al. (Graebner et al., 2012) proposed a system that performs the classification of customer reviews of hotels by means of a sentiment analysis. They used a corpus to extract the domain specific lexicon to be used in classification and classified reviews as positive or negative. Taboada et al. (Taboada et al., 2011) took advantage of linguistic resources like dictionaries and built a sentiment analyzer named SO-CAL that was similar to the work done by Polanyi and Zaenen (Polanyi et al., 2006).

One of the main drawbacks in lexicon-based approaches is the lack of scalability. For solving this problem, Neviarouskaya and colleagues (Neviarouskaya et al., 2008) described methods to automatically generate and score a new sentiment lexicon, called SentiFul, and expanded it through direct synonymy relations and morphologic modifications with known lexical units. They used four types of affixes in their work in sentiment features: propagating, reversing, intensifying, and weakening. Qiu and colleagues (Qiu et al., 2009) proposed a novel propagation approach that exploits the relations between sentiment words and topics or product features that the sentiment words modify, and also sentiment words and product features themselves to extract new sentiment words.

SentiWordNet is a known resource in sentiment analysis. Ohana and Tierney (Ohana et al., 2009) used the polarity values of words in SentiWordNet to classify movie reviews as positive or negative. In essence, their approach was simple in that they counted the polarity scores of polar words and then improved the approach by adding new features like negation to it. They also used machine learning techniques for classification. This work is explained more in Section V-D because it is the closest one to our approach.



For classifying the product reviews as positive or negative, existing techniques utilize a list of opinion words. Ding and colleagues (Ding et al., 2008) proposed a holistic lexicon-based approach to increase the accuracy of opinion mining tasks by exploiting external evidences and linguistic conventions of natural language expressions.

Another work similar to ours is proposed by Hamouda and Rohaim (Haouda et al., 2011). They obtained the polarities of the words inside a document from SentiWordNet and classified reviews as positive or negative based on the summation and average of those polarity scores. They also tried several values as threshold for distinguishing subjective and objective words. Their classification accuracy was around 69% in the best case. Finally Kaji and Kitsuregawa (Kaji et al., 2007) used structural clues that could extract polar sentences from Japanese HTML documents, and built lexicon from the extracted polar sentences. The key idea was to develop the structural clues. This work was able to provide high precision but not high recall.

## 5.2 Subjectivity Based Feature Extraction

In supervised training based approaches, sample reviews with known sentiments are used for training a classifier to distinguish between positive and negative reviews, considering the extracted features. Then, given a sample review in testing phase, the same features are extracted and compared to the learned models of positive and negative reviews. We use the average polarities and weighted polarities of different parts of the review as features, as summarized in Table 16. The first five features are computed using word polarities obtained from SentiWordNet, while the last five features are computed using the word polarities obtained from SubjWords.

**Table 16. Features extracted for each review**

Feature type	Feature name
<b>Domain-independent (Using SentiWordNet)</b>	F1: Average polarity of all words F2: Average polarity of negative words F3: Average polarity of positive words F4: Average polarity of last 3 sentences F5: Average polarity of first 3 sentences
<b>Domain-specific (Using SubjWords)</b>	F6: Cumulative frequency of positive words F7: Cumulative frequency of negative words F8: Proportion of positive to negative words F9: Weighted probability of positive words F10: Weighted probability of negative words

### Features Based on SentiWordNet (F1-F5):

In SentiWordNet, three scores are assigned to each connotation of a word: positivity, negativity and objectivity (Baccianella et al., 2010). The summation of these three scores equals to one:

$$Pos.Score(w) + Neg.Score(w) + Obj.Score(w) = 1$$

where  $w$  stands for a given word; and the three scores stand for its positivity, negativity and objectivity scores, respectively.

Furthermore, we define the the polarity of a word  $w$  as:

$$Pol(w) = Pos.Score(w) - Neg.Score(w)$$

We only consider adjectives and adverbs in a review since they are the most informative terms for sentiment analysis. As a preprocessing step, we eliminated all the words except for the adjectives and adverbs from the reviews. Therefore a word  $w_i$  in  $r$  denotes an adjective or an adverb in  $r$ . We also do not do word sense disambiguation and use the average polarity of all senses of a word. However, we include all the senses indicated by the POS tag of the word in the context, i.e. if a word is marked as adjective in a sentence, we use only the

adjective senses of the word and compute their average over the adjective senses of the word. Then, the average polarity of all words in a review,  $r$ , denoted by  $AP(r)$  is computed as in (1).

$$AP(r) = \frac{1}{|r|} \sum_{w_i \in r} Pol(w_i)$$

where  $|r|$  is the number of words -adjectives and adverbs- in review  $r$  and  $Pol(w_i)$  is the polarity of the word  $w_i$  as defined above.

The first three features (F1, F2, F3) are based on the average polarity concept (AP): F1 computes the average polarity of all words and F2 and F3 compute the average polarity of only the negative and positive words in a review, respectively. A word  $w$  is decided as positive if  $Pol(w) > 0$ , and decided as negative otherwise.

Usually authors express their opinion more directly in first or last parts of a review. In order to factor this information, we used two features (F4, F5) as the average polarity of words in last and first three sentences of a review. The features in this section are domain-independent because we extract the polarity of adjectives and adverbs from SentiWordNet which is a domain-independent polarity lexicon.

### Features Based on SubjWords (F6- F10)

Initial Seed Words includes 2005 positive, and 4783 negative words, which is filtered to construct a domain dependent set of SubjWords. This way we select significantly subjective words for a given domain.

Specifically, we construct the SubjWords from the Initial Seed Words based on their occurrence in the training set of labeled reviews which we call AdaptationReviews. The set of AdaptationReviews was also used to calculate the probability distributions for features F9 and F10 to train classifiers. Since evaluation was done using cross-validation on the training set, the AdaptationReviews is selected to be a completely different set of reviews to prevent biased testing. We select a word from the InitialSeedWords to be included in SubjWords if it appears in the set of AdaptationReviews. Our motivation behind this selection is that if a positive word appears in a significant number positive hotel reviews, most probably it will appear among other positive hotel reviews as well. The same argument holds for the negative words. We denote the final selection of positive seed words in SubjWords as PS and the final selection of negative seed words in SubjWords as NS for the formulation of the subjectivity-based features.

In sentiment analysis, seed word sets are often used by taking into account their occurrences in a review. An alternative is to use measures such as  $tf \cdot idf$  (term frequency\*inverse document frequency) (Martineau et al., 2009). The features F6 and F7 in our work are based on term frequency values, while F8 through F10 are the newly proposed features based on the occurrence of subjective words in SubjWords. Specifically, for F6 and F7 we compute the cumulative term frequency of positive and negative seed words for each document in the training set, respectively.

$$F_6(r) = \sum_{t_i \in PS} tf(t_i, r)$$

$$F_7(r) = \sum_{t_i \in NS} tf(t_i, r)$$

Here,  $F_6(r)$  is the cumulative frequency of positive seed words in review  $r$ ;  $tf(t_i, r)$  is the frequency of term  $t_i$  in review  $r$ . Similarly  $F_7(r)$  is the cumulative frequency of negative seed words in review  $r$ . Because usually negative reviews are dominated by positive ones and a considerable number of misclassified reviews are negative reviews that have been misclassified as positive, we increased the weight of the negative words by multiplying their frequency by 2; obtaining some improvement in accuracy according to the experimental results.

Since F6 and F7 give information about positive or negative term frequencies, we added feature F8 which is the proportion of positive seed words(the number of occurrences) to the negative ones in a review:

$$F_8(r) = \frac{p + 1}{n + 1}$$

$F_8(r)$  is the proportion of number of positive terms to negative ones in review  $r$ ; and  $p$  and  $n$  are the number of positive and negative seed words, respectively.

Finally features F9 and F10 are the weighted probabilities of positive and negative words in a review, calculated as follows:

$$F_9(r) = p * (1 - P_+(p))$$

$$F_{10}(r) = n * (1 - P_-(p))$$

where  $F_9(r)$  is the weighted probability of positive words in a review  $r$ ;  $p$  is the number of positive seed words in  $r$  and  $P_+(p)$  is the probability of seeing  $p$  positive words in a review. Similarly,  $F_{10}(r)$  is the weighted probability of negative words in a review  $r$ ;  $n$  is the number of negative seed words in the review, and  $P_-(n)$  is the probability of seeing  $n$  negative words in a review. Probabilities  $P_+(p)$  and  $P_-(n)$  are calculated from the set of AdaptationReviews.

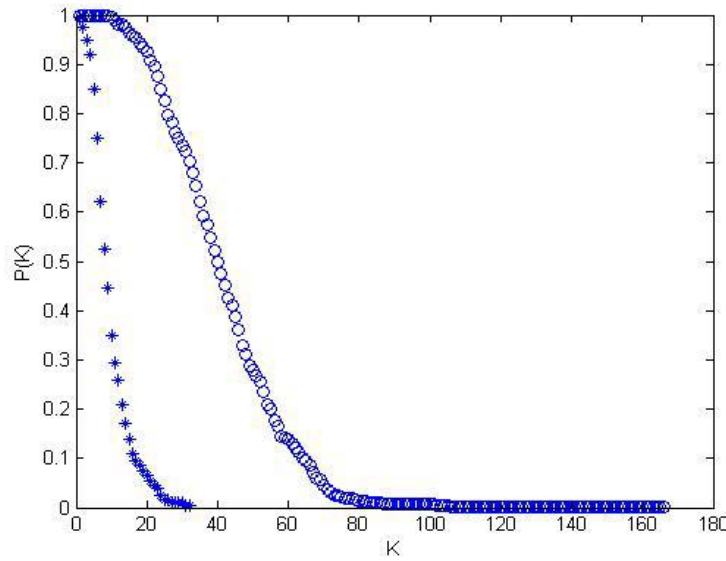


Fig. 8. Plot of  $P_+(p)$  as a function of  $p$  (\*' represents the hotel and 'o' represents the movie domain).

Note that  $(1 - P_+(p))$  increases as  $P_+(p)$  decreases; hence, we assign a large weight to an unlikely event such as the occurrence of a very high number of positive words that has a low probability; similarly for  $P_-(-)$ . Also, while it may seem like  $P_+(p)$  and  $P_-(n)$  would result in simpler features, the number of positive or negative words, are useful to enhance the feature especially for similar values of  $P_+(p)$  (e.g. for  $k \in [25 - 35]$  in the hotel domain).

The two probability distributions  $P_+(k)$  and  $P_-(k)$  are learned over the training set, separately for the two domains. The plot of  $P_+(p)$  is displayed for various  $p$  values, in Fig. 1. For instance 85% of all hotel reviews contain at least 5 positive words; or correspondingly,  $P(5) = 0.85$ . Note that the movie reviews are longer, and hence they contain more positive and negative words on average and the probabilities drop more slowly.

### 5.3 Experimental Evaluation

We used three widely used classifiers: Neural Networks, Logistic Regression and Support Vector Machines (SVM) within the Weka 3.6 software. Parameters of these classifiers are as follows. In the SVM classifier, we set  $\gamma$  to 0, loss to 0.1,  $C$  to 0.001 and cost to 1.0. In Neural Networks, the learning rate was set to 0.3, the number of hidden layers was set to 1 and the validation threshold was set to 20.

We used the hotel reviews from TripAdvisor corpus and movie reviews from Movie corpus (Pang, 2004). The TripAdvisor corpus consists of approximately 250000 customer-supplied reviews of 1850 hotels and was provided by (Wang et. al., 2010). We evaluated the performance of our approach on a randomly chosen subset of the corpus for simplicity. The randomly selected hotel review dataset consists of 6000 reviews half of which are positive, and the other half are negative. We assumed the hotel reviews with rating 1 or 2 as negative, and reviews with 4 or 5 star rating as positive for training. We did not consider the 3 star rating reviews since they do not convey a strong sentiment. We used all reviews in the movie set introduced by [16], including 1000 positive and 1000 negative reviews.

### Construction of SubjWords for the hotel and movie domains

The domain independent set of InitialSeedWords includes 2005 positive, and 4783 negative words. The set AdaptationReviews are randomly selected among the reviews from the hotel and movie domains, and it contains 1000 reviews for hotel and 1000 reviews for movie domains. Consequently we were able to construct the domain dependent set of SubjWords consisting of 671 positive and 1393 negative words for the hotel domain and another set of SubjWords consisting of 1093 positive and 1977 negative words for the movie domain.

### Results

We did a comparative evaluation of the subjectivity-based features by evaluating the effectiveness of different feature sets provided in Table 16. The results are displayed in Table 17, grouped according to basic feature groups in increasing effectiveness and different feature subsets. When the contribution of some particular feature(s) is tested, this is indicated by '+' (e.g. F6-F7 + F10) and positive contributions are highlighted in bold.

We can make several observations regarding the results shown in Table 17:

- the best feature group in isolation is based on cumulative term frequencies (F6 and F7).
- the accuracy of domain-specific features, F6-F10 is better than the accuracy of domain-independent ones, F1-F5.
- the most useful addition is the positive to negative word ratio (F8) which is mostly positive. in both domains the best results are obtained using all features, except for one experimental setup (the accuracy of the SVM in the movie domain is highest using only F6 and F7, which may be due to suboptimal parameter optimization in SVMs).

**Table 17. The accuracy (%) of three classifiers on hotel and movie reviews**

Domain		Feature Subset Accuracy (%)		
		(SVM)	(NN)	(Logistic)
<b>Domain (1)</b>	Basic:F1-F5	81.58	81.24	81.47
	Pos/Neg. Ratio:F8	83.37	82.78	82.21
	Weight. Pol.: F9, F10	84.45	83.08	82.99
	Cumul. TF.: F6, F7	83.56	84.15	83.07
	Hotel F1-F5 + F8	86.36	86.80	86.10
	F6-F7 + F8	84.52	84.51	83.43
	F8 + F9-F10	85.07	83.48	82.48
	F6-F7 + F8-F10	84.50	84.39	83.02
	All: F1-F5 + F6-F10	87.10	87.08	87.51
<b>Domain (2)</b>	Basic:F1-F5	62.60	62.00	64.2
	Pos-Neg. Ratio: F8	67.95	67.50	68.30
	Weight. Pol.: F9, F10	69.25	65.85	65.75
	Cumul. TF.: F6, F7	70.65	70.25	71.05

	F1-F5 + F8	69.10	67.50	70.45
	F6-F7 + F8	67.20	71.25	72.25
	F8 + F9-F10	70.30	70.15	70.80
	F6-F7 + F8-F10	68.80	70.95	72.75
	All: F1-F5 + F6-F10	68.45	71.65	72.85

Comparing the cumulative polarity (F6, F7) and weighted polarity (F9, F10) features, we observe that they both work well with accuracy of around 83% in hotel domain and the accuracy between 65% and 71% in the movie domain. However, F6 and F7 features give a little higher accuracy than F9 and F10. Indeed the tf measure on an appropriate set of seed words usually gives good results in opinion mining applications.

The effect of feature F8 is also good on classification. It alone gave the accuracy of 83% in hotel and 68% in movie domains, which are is similar to the results obtained by features in the domain-specific group. Furthermore, adding F8 to other feature groups contributed positively. Finally, the set of all features works better than other sets. Hence, we conclude that although most of the features are dependent, each one carries some information that cannot be isolated from the others.

## Discussion

Some studies in the literature have used SentiWordNet for sentiment classification. The first five features (domainindependent features) can be seen in several works (Baccianella et al., 2010) and (Hamouda et al., 2011). However, to the best of our knowledge the domainspecific features, specially F<sub>9</sub> and F<sub>10</sub>, have not been used before. Most similar work to ours (Ohana et al., 2009). where authors worked on movie reviews. What Ohana and Tierney did in their work, is using SentiWordNet for sentiment classification of movie reviews. Briefly, their approach is based on extracting some features from SentiWordNet. They used an SVM for classification and their best accuracy is reported as 69.35% (TABLE 18). Our approach is similar to theirs but our feature set, specially the second part of features using domain-specific polarity lexicon, is completely different.

**Table 18. Comparison of the accuracy(%) of two approaches**

Approach Data Set	<b>Our Approach</b> (Hotel)	<b>Our Approach</b> (Movie)	Ohana [4] (Movie)
Accuracy	<b>87.51</b>	<b>72.85</b>	69.35

## 6. Learning Domain-Specific Polarity Lexicons

Polarity lexicons, often used in sentiment analysis, indicate how positive or negative each term in the lexicon is. However, since creating domain-specific polarity lexicons is expensive and time consuming, researchers often use a general purpose or domain independent lexicon. In this s, we address the problem of adapting a general purpose polarity lexicon to a specific domain and propose a simple yet effective adaptation algorithm.

The basic idea for domain adaptation is to learn the domain specific polarities from labeled reviews in a given domain. In order to do that, we analyze the occurrence of the words in the lexicon in positive and negative reviews in a given domain. If a particular word occurs significantly more in positive reviews than in negative reviews, then we assume that this word should have positive polarity for this domain, and vice versa. We propose a couple of alternatives for the update mechanism of a word's polarity. The proposed approaches allow us to adapt a domain-independent lexicon such as SentiWordNet, for a specific domain by updating the polarities of only a small subset of the words. However, we also show that this small set of updated words has a significant contribution to sentiment analysis accuracy.

### 6.1 Introduction

The problem of identifying the polarity of words have been addressed in (Hatzivassiloglou et. al., 1997) where authors apply a clustering method to determine the polarity of adjectives. Similarly authors in (Wiebe et. al., 2000) use a set of seed words and clustering methods to find the polarity of adjectives in a corpus. More recently, polarity lexicons, such as the SentiWordnet, have been built for sentiment analysis. A polarity lexicon can be constructed manually (Das et. al., 2001), using heuristics (Kim et. al., 2004), (Baccianella et. al., 2010) or by machine learning techniques (Turney et. al., 2002). In (Liu et. al., 2012), they discuss three main approaches for opinion lexicon building: manual approach, dictionary-based approach, and corpus-based approach. The major shortcoming of the manual approach is the cost (time and effort) to hand select words to build such a lexicon. There is also the possibility of missing important words that could be captured with automatic methods. Dictionary-based approaches work by expanding a small set of seed opinion words, with the use of a lexical resource such as the WordNet (Fellbaum et. al., 1998). The main drawback of these approaches is that the resulting lexicon is not domain specific. Corpus-based approaches can overcome these problems by learning a domain-specific lexicon using a domain corpus of labeled reviews. The most commonly used polarity lexicon is the SentiWord- Net, where a word is associated with a negative polarity to indicate its negative sentiment orientation, a positive polarity to indicate its positive sentiment orientation, and an objective polarity to indicate its neutrality. The basic assumption with the polarity lexicons is that a word's polarity is the same across different domains, which may not be true for all the words.

For example, the word "small" has a polarity of -0.25 in SentiWordNet which is appropriate in the hotel domain like in the review sentence "The hotel had really small rooms". However, when the review is about a digital camera, the word "small" should have a positive polarity as in the review sentence "This camera is great as it has a small size".

The idea of updating the polarities of words in a given lexicon has been investigated before. In (Wilson et. al., 2009) authors stress the importance of contextual polarity to differentiate from the prior polarity of a word. They extract contextual polarities by defining several contextual features. In (Qiu et. al., 2009), a double propagation method is used to extract both sentiment words and features, combined with a polarity assignment method starting with a seed set of words. In (Choi et. al., 2009), which has been the main motivation for this work, authors use linear programming to update the polarity of words based on specified hard or soft constraints. For instance, if a word has negative polarity in the domain-independent lexicon but appears together with positive words in general, then its polarity is updated to positive, to minimize the costs imposed by the soft constraints. Another application of linear programming appears in (Dragut et. al., 2010). to learn a sentiment lexicon which is not only domain specific but also aspect-dependent. Lastly, a recent work expands a given dictionary of words with known polarities by first producing a new set of synonyms with polarities and using these to further deduce the polarities of other words (Lu et. al., 2011).

In this work we propose several variations of a simple method which is based on the delta tf-idf concept (Martineau et. al., 2009), to adapt a domain-independent polarity lexicon to a specific domain. We use SentiWordNet, a domain independent lexicon, as a baseline for demonstrating the effectiveness of the proposed method. Specifically, we adapt SentiWordNet to two different domains in order to obtain two domain-specific

lexicons. We then compare the sentiment classification accuracies obtained with SentiWordNet and the new domain specific lexicons.

## 6.2 Sentiment Analysis with Domain Independent Lexicon

We show the advantages of adapting a domain-independent polarity lexicon by comparing two approaches: (1) sentiment analysis using a domain-independent lexicon as explained in this section, and (2) sentiment analysis using the adapted lexicon exactly like the first approach. The adaptation process is explained in Section III.

The polarity lexicon we use as the domain-independent lexicon is the SentiWordNet that consists of a list of words with their POS tags and three associated polarity scores  $\langle pol^-, pol^=, pol^+ \rangle$  for each word [3]. The polarity scores indicate the measure of negativity, objectivity and positivity, and they sum up to 1. Some sample scores are provided in Table 19 from SentiWordNet.

**Table 19 Sample entries from SENTIWORDNET**

Word	Type	Negative	Objective	Positive
sufficient	JJ	0.75	0.125	0.125
comfy	JJ	0.75	0.25	0.0
moldy	JJ	0.375	0.625	0.0
joke	NN	0.19	0.28	0.53
fireplace	NN	0.0	1.0	0.0
failed	VBD	0.28	0.72	0.0

### Word Polarity

As many other researchers have done, we simply select the dominant polarity of a word as its polarity and use the sign to indicate the polarity<sup>direction</sup>. The dominant polarity of a word  $w$ , denoted by  $Pol(w)$ , is calculated as:

$$Pol(w) = \begin{cases} 0 & \text{if } \max(pol^-, pol^=, pol^+) = pol^= \\ pol^+ & \text{else if } pol^+ > pol^- \\ -pol^- & \text{otherwise} \end{cases}$$

In other words, given the polarity triplet  $\langle pol^-, pol^=, pol^+ \rangle$  for a word  $w$ , if the objective polarity is the maximum of the polarity scores, then the dominant polarity is 0. Otherwise, the dominant polarity is the maximum of the positive and negative polarity scores where  $pol^-$  becomes  $-pol^-$  in the average polarity calculation. For example, the polarity triplet of the word "sufficient" is  $\langle 0.75, 0.125, 0.125 \rangle$ ; hence  $Pol(\text{"sufficient"}) = -0.75$ . Similarly, the polarity triplet of the word "moldy" is  $\langle 0.375, 0.625, 0.0 \rangle$ ; hence  $Pol(\text{"moldy"}) = 0$ . An alternative way for calculating dominant polarity could be to completely ignore the objective polarity  $pol^=$  and determine the  $Pol(w)$  of the word to be the maximum of  $pol^-$  and  $pol^+$ . With this method, the dominant polarity of the word "moldy" would be  $-0.375$  instead of 0. However, we preferred the first approach as more appropriate, since many words appear as objective or dominantly objective in SentiWordNet.

### Review polarity

For estimating the sentiment in a review, we use a simple approach that computes the average review polarity and makes a decision based on this score. This is done by first applying the Stanford NLP tool to all the reviews in order to extract the POS tags of each word. Then, we compute the average polarity of the review using the dominant polarity of each word in the review using Eq. 2, using only words with POS tags JJ\*(Adjective), RB\*(Adverb), NN\*(Noun), and VB\*(Verb) which have dominant polarity positive or negative (we do not count the objective polarity words as their dominant polarity is 0).

$$\text{Average review polarity}(R) = \frac{1}{|R|} \sum_{w_i \in R} Pol(w_i)$$

The reviews with average polarity score greater than a threshold of zero are classified as Positive, while others are classified as Negative (zero score is classified as Negative). This threshold was found experimentally to give a roughly equal number of mistakes in positive and negative review classification.

### 6.3 Adapting Domain Independent Lexicon

Our purpose is to update the polarities of words in a given polarity lexicon to adapt them to specific domains. In this section we describe our approach for domain adaptation together with the methods we used for selecting the set of adapted words for sentiment classification.

#### Finding domain specific words

For adapting the general purpose lexicon, we update the polarity of a word, if its occurrence in labeled reviews strongly indicate one class, while SentiWordNet would suggest the other class. For instance if a word's dominant polarity is negative, but it occurs very often in positive reviews and not very often in negative ones, we update its dominant polarity.

In order to see which words in the domain appear more in a particular class of reviews, compared to the other class, we first compute the tf-idf (term frequency - inverse documentfrequency) scores of each word separately for positive and negative review classes. The  $tf(w; c)$  counts the occurrence of word  $w$  in class  $c$ , while  $idf(w)$  is the proportion of documents where the word  $w$  occurs, discounting very frequently occurring words in the whole database (e.g. 'not', 'be') (Salton et. al., 1975). There are quite a few variants of tf-idf computations (Paltoglou et. al., 2010), and the tf-idf variant we use is denoted as  $tf:idf$  and computed as:

$$tf * idf(w_i, +) = \log_e(tf(w_i, +) + 1) * \log_{10} \left( \frac{N}{df(w_i)} \right)$$

$$tf * idf(w_i, -) = \log_e(tf(w_i, -) + 1) * \log_{10} \left( \frac{N}{df(w_i)} \right)$$

where the first term is the scaled term frequency (tf) and the second term is the scaled inverse document frequency (idf). The term  $df(w_i)$  indicates the document frequency which is the number of documents in which  $w_i$  occurs and  $N$  is the total number of documents (reviews in our case) in the database. In Eq. 4, we define a new measure for polarity adaptation of words, called  $(\Delta tf)idf$ . It estimates whether the polarity of a word should be adjusted, considering its occurrence in positive and negative reviews separately.

$$(\Delta tf)idf(w_i) = tf * idf(w_i, +) - tf * idf(w_i, -)$$

Our new measure is similar to the *Delta TFIDF* term defined in [14] for calculating the polarity scores of words. As shown in Eq. 5,  $\Delta TFIDF(w_i; d)$  score of a word  $w_i$  in document  $d$  considers the difference in the document frequencies of that word in positive and negative corpora. Then, these scores are summed for each word in document  $d$ , to obtain a sentiment value for the document. In contrast,  $(\Delta tf)idf(w_i)$  of word  $w_i$  considers the difference between the *term* frequencies of the word  $w_i$  in positive and negative reviews.

$$\Delta TFIDF(w_i, d) = tf(w_i, d) * [idf(w_i, +) - idf(w_i, -)]$$

In this process we excluded words with POS tags containing "PRP" or "DT" to exclude stop words such as "the", "I", "a", etc. A portion of the resulting features are shown in Table 20.



## Updating word polarities

When we observe a disagreement between the SentiWord-Net polarity and the  $(\Delta tf)idf$  score of a word, we consider changing its polarity. For instance in Table 20, the word "joke" has a positive polarity, while its  $(\Delta tf)idf$  score is negative, indicating that it occurs more in negative reviews. Similarly, the word "comfy" has a negative polarity, while its  $(\Delta tf)idf$  score is positive, indicating that it occurs more in positive reviews.

For deciding on the new polarity of a word where a mismatch is observed, there are a couple of alternatives:

- *Flip*: Using the opposite polarity of the word (if the negative polarity of a word was dominant, we switch to its positive polarity and vice versa)
- *ObjectiveFlip*: Switching the objective polarity words to either negative or positive; similarly switching the negative or positive word to objective instead of its opposite polarity as done in *Flip*.
- *Shift*: Shifting the polarity of a word toward the other pole (as in adverbs, (Li et. al., 2010), (Ikeda et. al., 2008).)
- *DeltaScore*: Computing the new polarity based on the  $(\Delta tf)idf$  score of the word.

We tried two of these combinations (*Flip* and *DeltaScore*) and report results in large databases in two separate domains, as described in Section 4. As can be seen in Table below, in our experiments we observed that *Flip* has updated the polarity of the word "joke" in the TripAdvisor dataset. The SentiWordNet dominant polarity of the word "joke" was +0.53 and the updated polarity is -0.41. Indeed, the word "joke" appeared in a sentences like "Check in was a joke,...." where it contributes to negative sentiment.

## Extent of the Updates

For choosing how many words to update, there can be a couple of different alternatives:

- *Top-k%*: changing the polarity of the top-k% of the words showing a mismatch. For this option, we ranked the words with respect to decreasing  $j(\Delta(tf))idf$  scores and examined the top-k% of the list for sign disagreements between the  $(\Delta tf)idf$  scores and the SentiWordNet polarity.
- *Threshold*: changing the polarity of all the words below/above a fixed threshold where a disagreement occurs (e.g.  $(\Delta tf)idf < -10$  OR  $(\Delta tf)idf > 10$ )
- *Iterative*: changing the polarity of a word one at a time using hill-climbing, where the change would be accepted if it was found to improve accuracy on the validation set.

We tried all three approaches, but report the first two as the third option is too slow and not better than the others. For the *Top-k%* selection, we tried top-5 and top-10%. For the *Threshold* selection, we tried two runs with different positive and negative threshold value ranges that will enable a good number of words to be picked. Notice that all of these update methods can also include the *ObjectiveFlip* approach where *Top-k%* would be modified to have a *Middle-k%* and *Threshold* would have two threshold values to determine the middle range of words such that  $2 < (\Delta tf)idf < -1$ .

## 6.4 Experimental Evaluation

We implemented and tested the proposed polarity adaptation techniques on real review data sets to assess their impact on the overall sentiment classification. Following sections detail the data sets used and the results obtained on these data sets.

### Data Sets

We tested our system in two different domains: hotel and movie reviews. For the hotel domain, we extracted 6000 reviews from a snapshot of the TripAdvisor site which was prepared by (Wang et. al., 2010). In order to have an equal number of positive and negative reviews, we randomly collected samples from this larger set, resulting in 1500 positive and 1500 negative reviews in Train and Test datasets. In the hotel reviews, a review may have a rating of 1,2,3,4,5 where we assume that the reviews with rating 1, and 2 are negative and the reviews with rating 4, and 5 are positive. We did not consider the reviews with rating 3 for training since they do not have a strong polarity. For the movie review domain, we use the movie review data provided by (Pang et.

al., 2004), consisting of 2000 reviews. In order to have an equal number of positive and negative reviews, we randomly split this database into two sets, Train and Test datasets, each containing 500 positive and 500 negative reviews. In the movie reviews, a review is marked with label ”-” to indicate that it is a negative review and marked with ”+” to indicate that it is a positive review.

## Results

The results for two datasets are shown in following Tables. We have tried Flip, and DeltaScore updating methods, with top-5% and top-10% of all the words. We also tried the Threshold update with different threshold values for picking the words to flip.

As can be seen in these tables, the updates all show improvement over the alternative of using SentiWordNet polarity values without doing any adaptation. These improvements are comparable to those in (Choi et. al., 2009) where around 2% accuracy had been obtained using an adaptation done by linear programming.

$(w_i)$			$idf(w_i)$	$\Delta tf(w_i)$	$pol(w_i)$	Result
treat	21	0	4.96	15.34	0.77	Agreement
exceeded	15	0	5.23	14.51	0	
neat	19	2	5.01	9.51	0.48	
dirty	19	249	2.65	-6.7	-0.47	
smelly	0	24	4.79	-15.41	-0.75	
fireplace	37	1	4.57	13.46	0	Disagreement
comfy	72	13	3.64	6.01	-0.75	
Pleased	41	11	4.09	5.13	-0.5	
Joke	5	40	4.29	-8.25	0.53	

**Example word flips from movie dataset**

Word	SentiWordNet	DeltaScore	Review	Context	Review Polarity
failed	0	-0.73	The speciality restaurants tried to be americanized but failed horribly.		2
joke	0.53	-0.41	Check in was a joke, our room ...		1
comfy	-0.75	0.30	The room was big enough with a large comfy bed.		5
sufficient	-0.75	0.49	Simple but sufficient complimentary breakfast (coffee, fruit, ...) left us satisfied.		4
treat	0	0.77	The lounge was a wonderful treat each morning and afternoon.		5
gem	0	0.81	What a gem!		5

**Example word flips from movie dataset**

Word	SentiWordNet	DeltaScore	Review	Context	Review Polarity
garbage	0	-0.47	... and i tend to shy away from watching such garbage	...	-
ludicrous	0.56	-0.36	... are so over the top, nonstop, and too ludicrous for words	...	-
implausible	0.44	-0.27	This movie was just completely implausible	...	-
laughable	0.56	-0.21	This is a laughable 1977 rip off of king kong (1976),	...	-
courage	-0.5	0.22	Few filmmakers have the courage and sheer audacity	...	+
complicated	-0.625	0.32	It is notable for introducing one of the first complicated gay characters...		+
jarring	-0.625	0.38	The tune is haunting, but it is also completely jarring.		+

## 7. Implementation and Evaluation on UBIPOL Data Sets

In this section we used real data set that was collected by UbiPOL partners. Our aim is to give some implementation details and to show our engine.

### 7.1 Opinion Mining Engine and its Screenshots

Its link: [ferrari.sabanciuniv.edu/sare](http://ferrari.sabanciuniv.edu/sare)

Step1: Home Page ([ferrari.sabanciuniv.edu/sare](http://ferrari.sabanciuniv.edu/sare))

**Add Files** button must be used to upload a Corpus. In this scenario Example comments (Ref: LBH file ) was shown in Table 7.1.

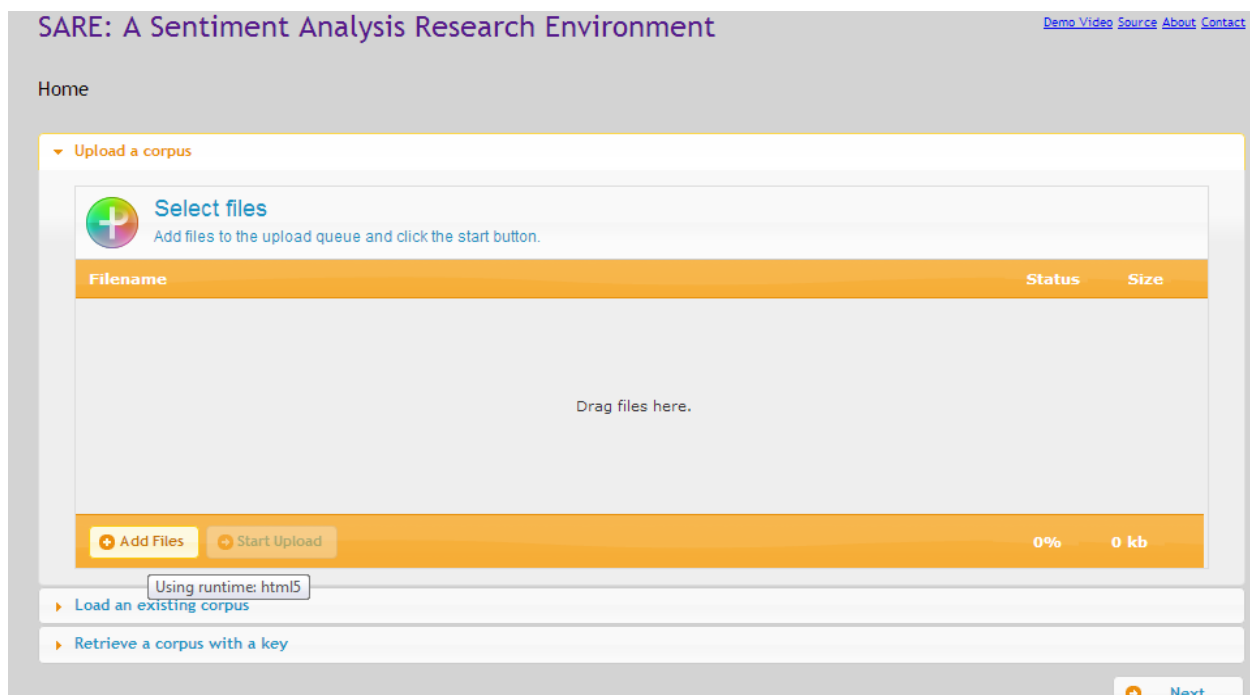


Figure 7.1 Home Page

Table 7.1: Comment Examples

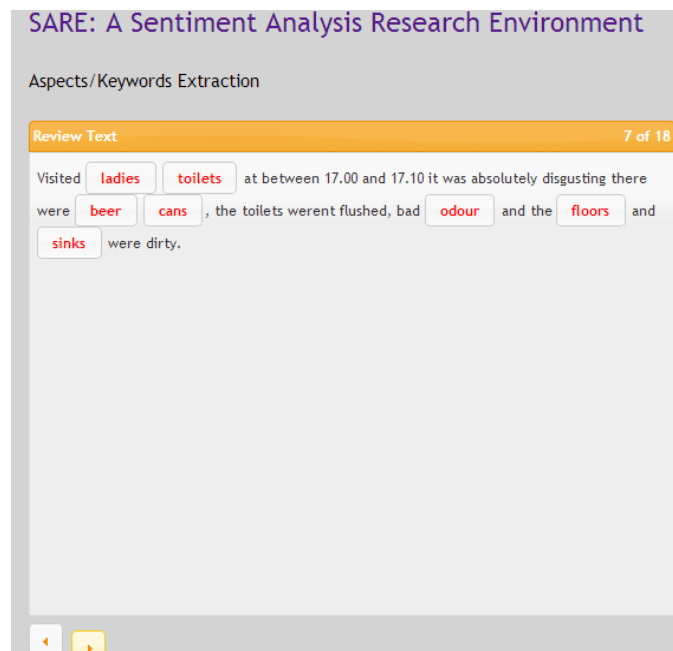
<p>I like your parking cards.</p> <p>There was no heating on the bus, spoke to driver and driver ignored passenger. Driver had a bad attitude.</p> <p>Quality services. Good communications. Value for money. Support for local issues.</p> <p>Caller is angry that no information in RTI told him.</p> <p>Rubbish collection is excellent, weekly collection, no wheelie bins, all recycling in one bag. Good libraries in Ruislip and little graffiti and streets are clean.</p> <p>Potholed roads - uneven pavements - messy refuse collection - half-hearted recycling policy - restrictive residents parking zones and times - detached Civic Centre 'call centre' - disinterested staff - officious operatives - pompous council officials.</p> <p>Services are efficient and effective. The area is well maintained and council support and advice seems to be available at most reasonable times.</p> <p>The new high street parking with 30 minutes free has gone a long way to solve parking problems.</p> <p>The resident discount card and free travel passes for pensioners are a great help. When the rest of the country seems to be besieged with rubbish collection problems Hillingdon provides an excellent continuous service.</p> <p>The tip is open good times our streets are clean.</p> <p>I would like to see more police and less parking wardens always asking opinions.</p>
---

Easy to contact the contact centree if necessary, good libraries.  
They deal with queries quickly on the phone and via email.  
They provide excellent parking charges with the hillindon resident card.  
Got on the slow train from Kirk Sandall which got into Doncaster, she swapped over to the fast train which should have left Doncaster at 7.  
I live in a small quiet street that is suffering serious road damage.  
I don't know if you know but the journey planner part of your website doesn't work properly, it brings up the right services but if at midday I put to arrive at 3.30pm.  
This bus stop pole has a damaged advert on the information carousel.  
Excellent sports facilities,waste collection, clean streets and green spaces.  
Council planning restricting back garden developement all together make Hillingdon Borough a pleasant place to live.  
Street cleaner comes regularly,bins are collected each week recycling is done.  
Lots of well kept greenspaces good healthcare facilities great bus and rail network visible policing good schools local forums  
The problem we have is connected to the armthorpe bus stop in the interchange in Doncaster, Nobody seems to know where or how to queue in an orderly fashion, the seats are too far away.  
Visited ladies toilets at between 17.00 and 17.10 it was absolutely disgusting there were beer cans, the toilets werent flushed, bad odour and the floors and sinks were dirty.  
The driver was swearing at the mans daughter for trying to get on the bus with a child over 11 without a MegaTravel pass.  
Arundel Gate - 37 bus - 10.13 -!!Same bus - last week - DID NOT COME!! Approx 5 weeks ago - same bus - 10.43 - DID NOT COME!!  
The driver was very helpful.

## Step2: Comment Screen

This screen displays single comment, press the right arrow to view next.

Figure 7.2 Comment Screen



## Step3: Domain ontology screen

Municipality ontology that is shown in Figure 7.3 was used. Domain knowledge contains set of aspect name and each aspect has set of keywords.

Figure 7.3 Aspect Keyword Set

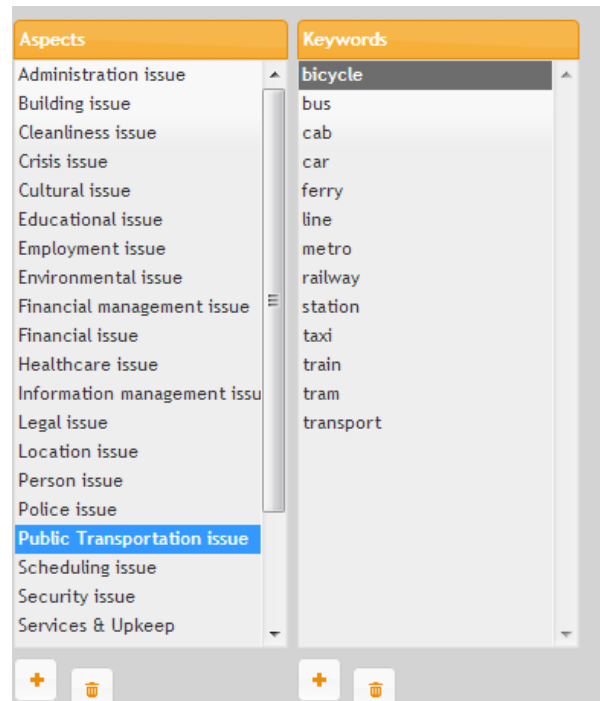


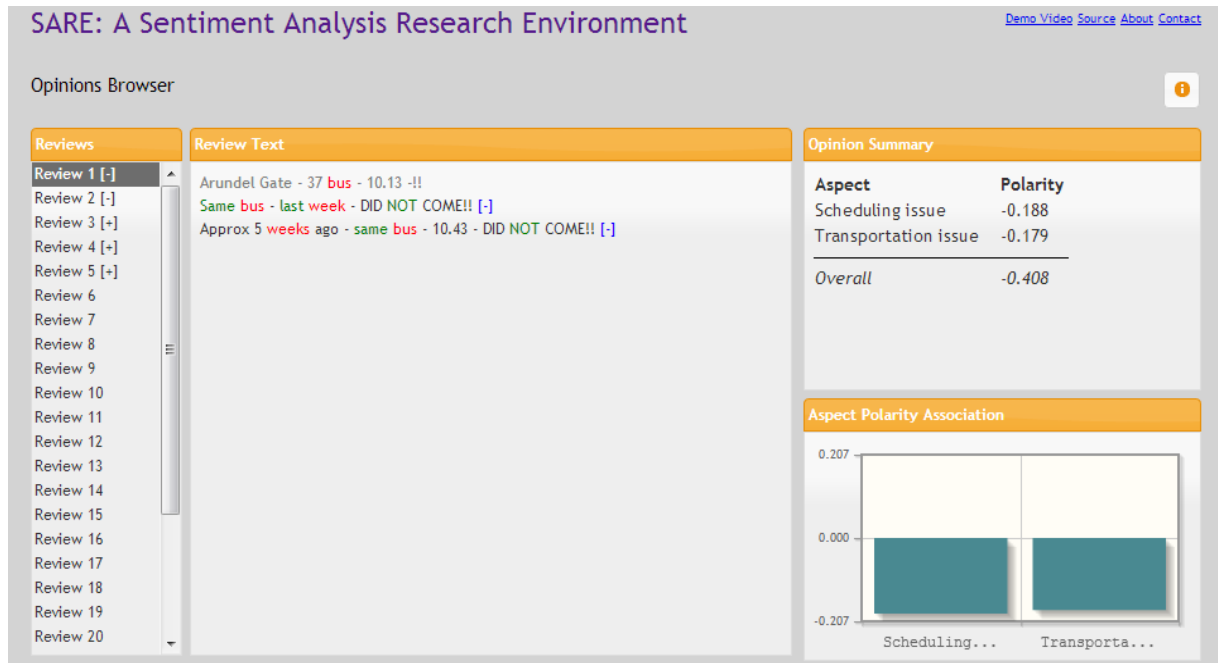
Table7.2:Aspect- Keyword Examples

Aspect Name	Keywords									
<Financial issue>	concession cut	costs	negligence	savings	budget	businesses	wastage			
	freezing									
	opportunities	crisis	discount	council tax	fare	charge	charges	value	rises	
	money	recession								
<Transportation issue>	parking zone	driveways	street	streets	garages	metro	tarmac	lumps	bridge	
	card	speeds	parking charge	pathways	pavement	train	Maintainence	train		
		surface								
	parking	wardens	restrictions	alleyway	eyesore	drivers	sliproad	tram	parking policy	
	traffic	motorbikes	pass	graveyards	road	potholes	network	pedestrian		
	concessions		road	property	lot	tarmac	car	roadwork	highway	
	station	one-way	name	pothole	grass	zone	transport	pathway	two-way	motorbike
	avenue	roundabouts		automobile	traffic	bus	terminal	dead	alleyway	hole
	bridge	public	pavement	train	warden	community	garage	lighting	route	
	parking	flowe								
<Healthcare issue>	end	address	exit	lights	traffic	snow	construction	bridge	jam	
	connection		ice	route						
<Social issue>	health	dentists	healthcare	insurance						
<Cultural issue>	charity	family	adults	childcare	children	age	community	healthcare		
	communication	music	book	press	websites	forums	arts	museum	art	
	library	media	events	online	information	magazines				
<Sport issue>	tennis	swimming	snooker	baseball	billiard	cricket	facilities	gymnastics		
		cycling	leisure	judo	basketball	badminton		pool		

## Step4: Result Screen

Review based results was shown at the left hand side.  
 Each review was analyzed sentence by sentence.  
 Opinion summary was shown at the right hand side.

Figure 7.4 Result Screen



## 7.2 Twitter Integration and Opinion Mining on Tweet Data

Home Page: [sky.sabanciuniv.edu/OpinionMining/index.html](http://sky.sabanciuniv.edu/OpinionMining/index.html)

Step1: For the municipality data, we select "Municipality" domain to analyze our data in this context.

Figure 7.5 Comment Screen

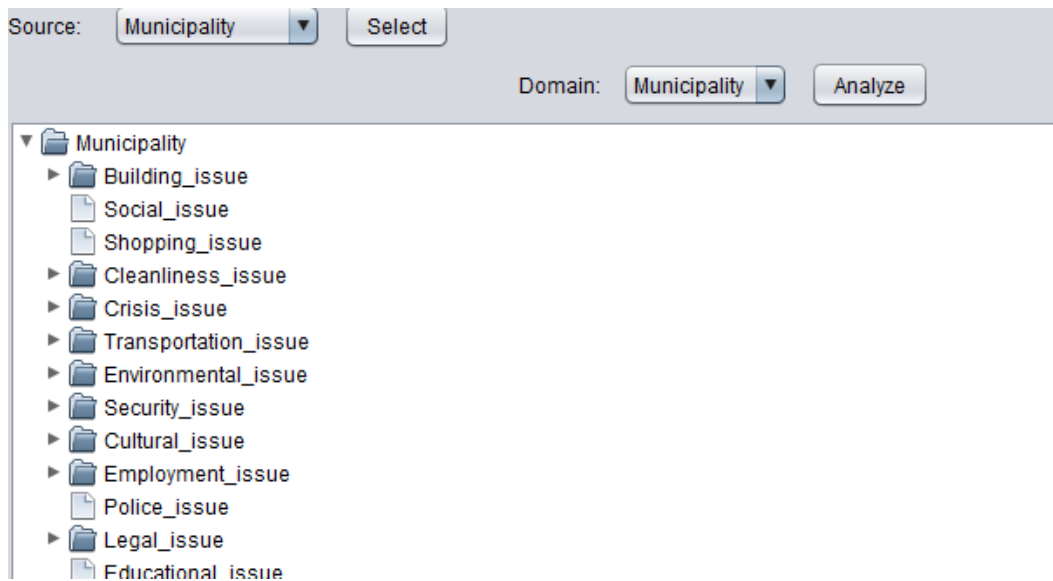
The screenshot shows a form for selecting data source and domain:

- Source:** A dropdown menu with "Municipality" selected and a "Select" button.
- Domain:** A dropdown menu with "Municipality" selected and an "Analyze" button.

Firstly, we select our file source where indicates we collect the data. By choosing "Municipality", instead we choose to read our data which is located in a specific file in our server. The next step is specifying domain information, such that we can work with different domains.

## Step2: Domain Ontology Screen

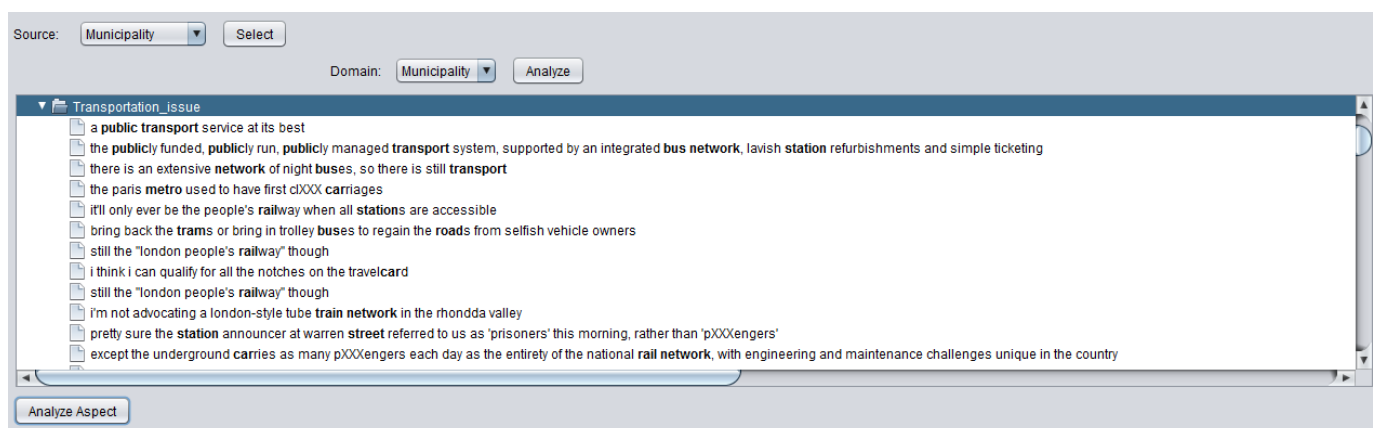
Figure 7.6 Municipality Ontology



We have an ontology file which we created before and it has different ontology concepts inside it. For each ontology, has a tree hierarchy such that each aspect has some number of keywords and also they may have sub-aspects as well. When we select domain information in our program, it will visualize the tree structure as it is shown in the window.

## Step3: Classification

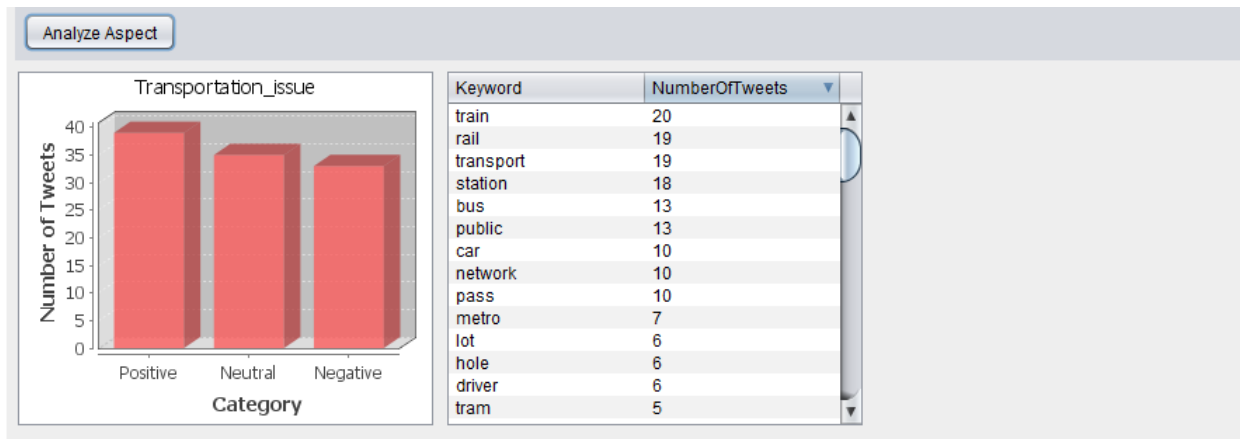
Figure 7.6 Aspect based Comment Categorization



Every sentence in this data is classified under one specific aspect according to the number of keywords it has. Additionally these sentences are ranked by a score value which is  $(\text{number\_Of\_Keyword\_In\_Sentence} / \text{number\_Of\_All\_Words\_In\_Sentence})$  and the keywords are represented as bold.

## Step:4 Result Screen

Figure 7.7 Opinion Mining



After we display sentence-aspect relations, we may use another feature of our program which is analyzing a specific aspect. More details, we are able to select a specific aspect and then press “Analyze Aspect” button which makes opinion mining on this selected aspect. In other words, we can visualize a graphical information which indicates the how many positive, negative and neutral sentences are used by users. Additionally, we can represent the keywords which are used under this aspect by sorting them according their frequency, and this feature also provides us to have idea about which keywords are frequently mentioned by users.

**Similar Process for the Turkish**

## Step1:

Source:

Domain:

## Step2:

Source:

Domain:

- ▼ ☐ Belediye
  - ▼ ☐ Belediye\_Aspects
    - ▶ ☐ Temel\_Hizmetler
    - ▶ ☐ Açık\_Alanlar
    - ▶ ☐ Yaya
    - ▶ ☐ Yan\_Hizmetler
    - ▶ ☐ Ekipman
    - ▶ ☐ Otopark
    - ▶ ☐ Kapalı\_Alanlar
    - ▶ ☐ Alışveriş\_Merkezi
    - ▶ ☐ Kişiler
    - ▶ ☐ Yol
    - ▶ ☐ Hava\_Koşulları



Step3:

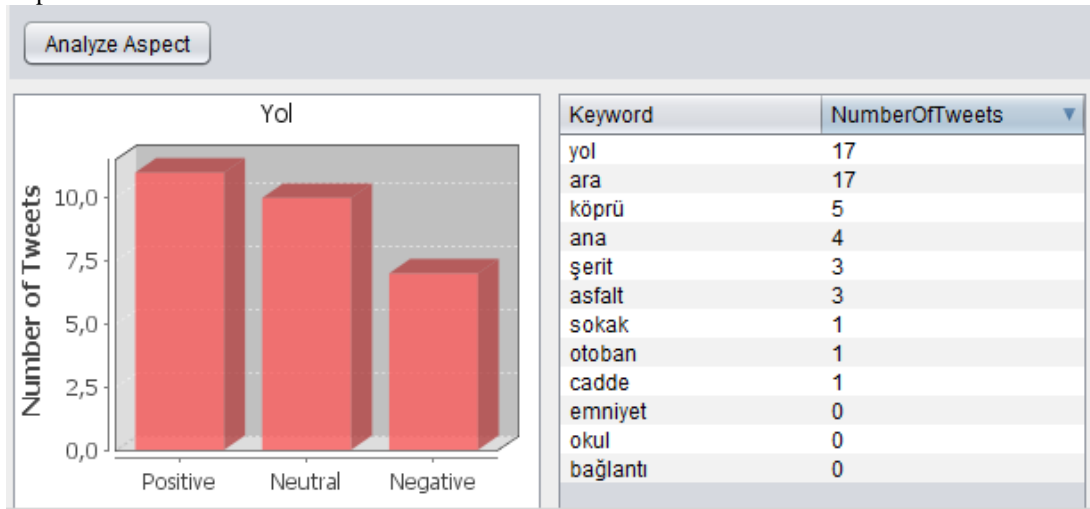
Source:

Domain:

▼ Yol

- ☐ ana yolların yeni **asfaltı** güzel
- ☐ \***ara** yollar çok yamalı
- ☐ \***anakra** trafiğinde herkesin sol **şeritten** gitme alışkanlığı trafikte sorunlara ve kazalara **yol** açmakta
- ☐ **zara** ilçesi tali yolların tümünü **asfalt** ve kaldırımları sıfırdan yaptırmıştır
- ☐ şehirler **arası** yollardaki kalite gün geçtikçe artıyor
- ☐ kış mevsiminde **yollara** dökülen tuz tüm türkiye'de **asfaltları** delik deşik ediyor
- ☐ **ankara'nın** her yerine metro şart
- ☐ zırlılırlıklarda **yol** yapım çalışmasından dolayı yeni yapılan **köprü** girişinde (tek **şerit** olduğu için) yoğunluk yaşanıyor
- ☐ "azıcık bir kar yağsa, istanbul trafiği kilitleniyor, özellikle **ana köprüler**
- ☐ şehir merkezlerinde bisiklet **yolları** yaygınlaştırılmalı sadece park içinde insanlar bisiklet kullanabiliyorlar
- ☐ ayrıca akşam etimesgut'a girişte turgut özal **köprüsü** girişinde 3 **şeritlik** **yol** 2 ye indirildiği için yine sıkışıklık yaşanıyor
- ☐ \*izmir de şehir içi neredeyse her yer **otoban** kalitesinde **yolları** ile birbirine bağlı
- ☐ **yol** tabelaları çok sayıda var ve başanlı

Step4:



### 7.3 End user Evaluation

We organized a meeting for the end use. The aim of the meeting was to obtain feedback from the Pendik and Yalova Municipalities as potential users of UbiPOL. We focused on the demonstration of the Reported Issues from the perspective of the Backend Application, and the Opinion Mining Engine.

#### 7.3.1 Backend Application Demonstration

We described the purpose of the meeting, made a brief presentation of UbiPOL, and a detailed demonstration of the backend application, the reported issue and their interaction. Also we informed the guests about how UbiPOL project is going to live beyond the projects timeframe and how the opinion mining part could be exploited by different branches of the municipality as part of UbiPOL framework.

- Members of the Yalova municipality have showed interest to participate as potential exploiters. Their opinion and questions about UbiPOL have been listed below:

- PM (Pendik Municipality) raised some concerns on the reported issues.
  - a- When the citizens create reported issues on the same topic, how are those reported issues handled, especially when the reported issues are very similar.
  - b- The time frame of the reported issue is important. For example, when an issue is reported, for how long can someone enter comments on the reported issue? When does the reported issue expire. In general time management needs special consideration.
- YM (Yalova Municipality) commented on the mobile application which are listed below:
  - a- There needs to be some authority who will be responsible for the announcement of the project and the system. Also the maintenance is very important. Who will be responsible for that and how? This needs to be clarified.
  - b- They showed great interest in using the mobile and backend application. They are curious about the procedures about how to deploy the system.
- Comments for the improvement of the backend application:
  - Municipality representatives indicated that the results of the questionnaires should be presented differently to the citizen and the head of the municipality. The head of the municipality needs to know exactly all the problems and the facts, however publishing the results of the questionnaires as they are may not be politically accepted. Therefore, when the system is deployed, this needs to be handled through filtering or special reporting tools.

### 7.3.2 Demonstration to Opinion Mining Engine

- We described the purpose of the sentiment analysis/opinion mining in general, and made a brief presentation of the UbiPOL opinion mining module. They did a detailed demonstration of how opinion mining is integrated into the backend application.
- As an example data set a municipality corpus was used together with opinion mining engine to explain how the opinion of the citizens can be extracted from textual documents.
- The idea behind the domain knowledge is explained in the context of municipalities. The concepts of aspect and keyword are explained on the opinion mining engine.
- A full demo of the opinion mining engine was done.

### 7.3.3 Twitter Extension

The twitter extension of the opinion mining engine was demonstrated to the municipalities which included the following functionalities:

- a- Online extraction of tweets about Pendik Municipality from their accounts using their account name and hashtags.
- b- Classification of the user tweets into aspects.

Their feedback and questions are as follows:

- PM (Pendik M.) about the citizen twitter comments:
  - c- What happens when there are insulting tweets? How do we eliminate them? However this is not very relevant for UbiPOL since the tweets are already public and can be seen by anyone. Therefore in the case of Twitter, this is not UbiPOL's problem. However, as an extension of UbiPOL, we may include insulting tweet detection in the future.
  - d- How do we construct ontology for the municipalities? SU asked if they have existing taxonomy for the tasks and responsibilities of the municipality. The municipalities indicated that they have some taxonomy which could be the bases of the ontology. They are going to share this taxonomy with UbiPOL.
- YM (Yalova M.) indicted that they are willing to use UbiPOL and the twitter extension as part of their citizen feedback collection mechanism.

- Comments for the improvement of UbiPOL System in general:
  - The system in general is very useful and user friendly, they want to stay in touch on the exploitation of UbiPOL system

**Also we arrange a Skype Meeting with ADA (29/03/2013)**

-The system in general is very useful and user friendly, they want to stay in touch on the exploitation of UbiPOL system

-Comments for the improvement of UbiPOL System in general:

- They asked a question to understand the structure of the reported issue title (neg or pos).
- We informed them that instead of the use subjective sentence, objective sentences must use.

## 8. Conclusion

---

In Section 2, we proposed a new anonymization model MSA-diversity which limits the probability of disclosure in datasets with multiple sensitive attributes. MSA-diversity is also resistant to attacks by adversaries with nonmembership information. We also present an anonymization algorithm which takes advantage of flexible hilbert space partitioning to ensure MSA-diversity. We experimentally show that, when compared to state-of-the-art algorithms, the MSA-diversity algorithm offers higher levels of privacy while achieving better utility in most practical privacy settings.

In Section 3, the proposed solution addresses the problem of privacy-preserving ranked multi-keyword search, where the database is outsourced to a semi-honest remote server. Our formal definitions pertaining to the privacy requirements of a secure search method are based on a comprehensive analysis of possible attack scenarios. One particular privacy issue concerning linking of queries featuring the identical search terms is often overlooked in literature. When an attacker is able to identify queries featuring the same search terms by inspecting the queries, their responses and database and search term statistics, he can mount successful attacks. Therefore, the proposed privacy-preserving search scheme essentially implements an efficient method to satisfy query unlink ability based on query and response randomization and cryptographic techniques. Query randomization cost is negligible for data controller and even less for the user. Response randomization, on the other hand, results in a communication overhead when the response to a query is returned to the user since some fake matches are included in the response. However, we show that the overhead can be minimized with the optimal choice of parameters. The true cost is due to the additional storage for extended index file and the actual search time.

This can also be minimized by proper selection of parameters (i.e., the ratio of fake index entries to real index entries). On the other hand, the storage is usually not a real concern for cloud computers considering that index file is relatively small compared to document sizes. As for the search time, the proposed technique is extremely efficient that a relative increase in search time can easily be tolerated. Our implementation results confirm this claim by demonstrating search time over a database of 10; 000 documents, including ranking, takes only a couple of milliseconds. Considering that the search algorithm easily yields to the most straightforward parallelization technique such as MapReduce, the overhead in search time due to the proposed randomization method effectively raises no difficulty.

Selection of parameters involves some knowledge about the database and therefore, an a priori analysis is required. However, our proposal needs only the frequency of the most used search terms and number of search terms used in queries. The formulation for parameter selection is simple and easy to calculate. Furthermore, we do not need to repeat the calculation process for different datasets. One can easily specify an upper bound on the frequency of the most used search terms and number of search terms that can be used for many cases. Ranking capability is incorporated to the scheme which enables the user to retrieve only the most relevant matches. The accuracy of the proposed ranking method is compared with a commonly used relevance calculation method where privacy is not an issue. The comparison shows that the proposed method is successful to return highly relevant documents. We implement the entire scheme and extensive experimental results using both real and synthetic datasets demonstrate the effectiveness and efficiency of our solution.

In Section 4, we tried to bridge the gap between word-level polarities and review level polarity through an intermediate step of sentence level analysis of the reviews. We formulated new features for sentence level sentiment analysis by an in-depth analysis of the sentences. We implemented the proposed features to see the effect of sentence level features on polarity classification. We observed that the sentence level features have an effect on sentiment classification, and therefore, we may conclude that sentences do matter in sentiment analysis and they need to be explored for larger and more diverse datasets such as blogs.

In Section 5, we worked on subjectivity-based features for sentiment classification. We used two lexicons for feature extraction and experimented on two different domains. We also proposed new subjectivity-based features which improved the sentiment classification accuracy. We used some ready resources: two sets of positive and negative words, used in domain-specific features and also SentiWordNet, used in domain-independent features. The efficiency of domain specific features was higher than domain-independent ones. After training and testing the system, we achieved an accuracy of about 87% in hotel and 72% in movie domains. One potential point of improvement of our approach could be in our classification of reviews into subjective and objective. Take for instance: “this hotel has body building area”. Although this sentence does not explicitly state any positive or negative connotation, it implies an advantage for the hotel. Having body building facility is a positive point for a

hotel. Also some idioms are actually subjective but difficult to distinguish: “staying in this hotel costs an arm and a leg”; which expresses a negative point for the hotel because it is too expensive.

As can be seen in the experiments section, our system with the newly proposed features obtains one of the best results obtained so far, except for (Bespalow et al., 2011). Its main drawback is that topic models learned by methods such as LDA requires re-training when a new topic comes. In contrast, our system uses word polarities; therefore it is very simple and fast.

In Section 6, we adapted an existing polarity lexicon to a specific domain by learning new polarity orientations for the words by looking at how they are used in a particular domain. Although the proposed method is very simple and efficient, the use of the resulting adapted lexicons have increased review sentiment classification accuracy in both of the tested domains.

In Section 7, we used end user comment corpuses to show our system’s output. First we showed opinion mining engine then explained tweet extension. Finally, we discussed with end users to get their feedbacks.

The next step of our investigation is to analyze and improve opinion mining engine effectiveness. We plan to test our approach with the trial related real life sample. Also we left a mechanism/procedure for online updating ontology as future work.