Identifying Time-varying Drivers of Social Media Issues and Conversations

Abstract

Successfully understanding social media conversation growth, dissemination and extinction is a challenging task that relies on identifying groups, group influence, diffusion models, forecast models, social dynamics and text analytics. In this problem, we concentrate on the description of a novel approach for identifying drivers of the direction and momentum of social conversations, including the spread of mood, sentiment and issues. The approach first groups potential drivers of conversation based on variability. The primary driver in each group is then selected. Finally, the relationship between the selected drivers and the topic outcome is calculated and displayed visually. This enables the quick identification of the form and structure of the conversation and allows us to predict momentum, direction, contagion risks, potential responses and interventions.

Context:

There is a huge amount of data in the form of text available today in the internet across various channels – social media, news articles, blogs, e-commerce websites. Most of this data is a part of some "conversation" or the other where real-world entities discuss, analyze, comment, exchange information in the form of written expressions in textual format. Driver Modeling on textual data can be useful in observing the key drivers which are driving the "conversation" coupled with the associated sentiments and mood states for the observed key drivers. These insights about the key conversational drivers are often used in a variety of domains such as tracking news cycles, stock movements, legislation developments, brand image, viral breakouts and much more

Keywords: Driver Analysis, Textual Topic, Latent Dirichlet Allocation, Sentiment Analysis, NLP, Sub-net Identification, Hypernyms Prediction

1. Introduction

In most e-commerce/connected commerce websites, the consumer base has the authority to comment or provide feedback pertaining to the services received. E-commerce/retail websites have the historical feedback/reviews received open for public viewing. 90% of consumers say buying decisions are influenced by online reviews; More than 50% of online purchasers said they had read online reviews prior to hitting the buy button, while nearly 40% of consumers who made purchases in-store did so *(Report published by Google in 2012).* For connected commerce business, ratings & reviews play a critical role in driving brand/product choice and ultimately conversion to purchase. There is an opportunity to mine the ratings & reviews data and convert it into a competitive advantage to drive conversion & loyalty, win vs. competitors, and ultimately drive incremental sales. There is an opportunity to mine the ratings & reviews data and convert it into a competitive advantage to drive conversion & loyalty, win vs. competitors, and ultimately drive incremental sales. The goal is to utilize technology (Natural Language Processing) in and machine learning to understand drivers of ratings that can then be leveraged to reach right consumers/right time/right place with right product at right price. Through NLP/ML, we would be able to better understand drivers of Ratings & Reviews, cluster consumers; deploy learnings across consumption generating vehicles and drive higher conversion/sales/loyalty. With over 60% of all outlet sales being digitally influenced, this would help drive both online and offline sales. Learnings will also influence innovation and communications strategy. The approach is illustrated in Figure 1 where we show the process flow.

1.1 Preliminary pre-processing

In order to perform Driver Analysis, it is necessary to process the natural language and convert it into a model-ingestible format. The major preliminary processes involved are as follows

- Stemming
- Lemmatization

- POS Tagging
- Dictionary and Taxonomy construction
- Acronym resolution
- N-gram Tokenization

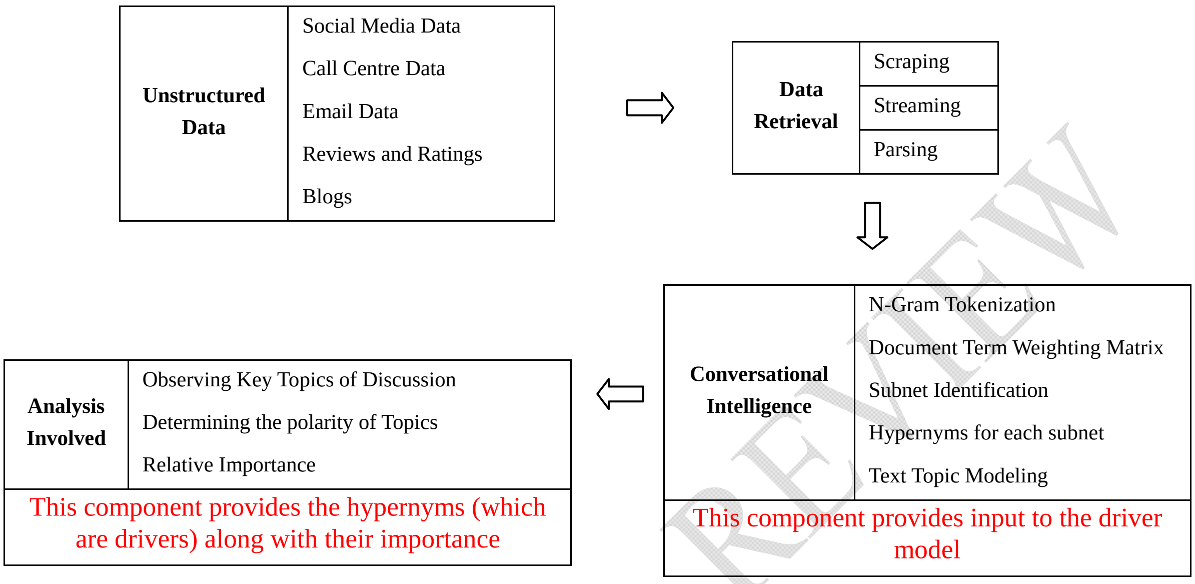

Unstructured Data
Social Media Data
Call Centre Data
Email Data
Reviews and Ratings
Blogs
Data Retrieval
Scraping
Streaming
Parsing
Conversational Intelligence
N-Gram Tokenization
Document Term Weighting Matrix
Subnet Identification
Hypernyms for each subnet
Text Topic Modeling
This component provides input to the driver model
Analysis Involved
Observing Key Topics of Discussion
Determining the polarity of Topics
Relative Importance
This component provides the hypernyms (which are drivers) along with their importance

Figure 1: Illustration of the high-level process flow

Unstructured Data	**Natural Language Text**	**Pre-processing**	**Tokenization**
	Totally love the product	Stemming & Lemmatization – to bring the words to its base form	Tokenization for the input of the topic model can be taken as n-grams, where n can be set by user as required
	Good for a gift set		
	…	Part of Speech Identification – to observe semantic context	
	A little expensive		

Figure 2: Preliminary pre-processing of textual data to make it ingestible

2. Key Components

The key functional unique components of the driver model are as follows:

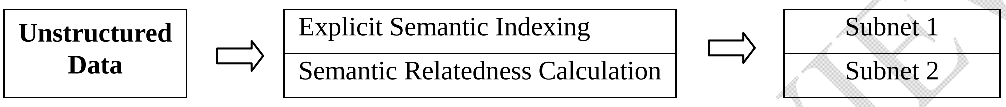
- **Community or Sub-net identification** and their respective hypernyms using ‘Relatedness’ matrix (Explicit Semantic Indexing)
- **N-gram Tokenization for Topic Modeling** - For each hypernym, we would be identifying the topics using n-grams
- **Weighing the Document-Term Matrix** with Inverse Document Frequency to provide higher weights to low-frequency terms and lower weights to high-frequency terms
- **Topic Modeling** on each subnet to identify top n-grams (hypernyms) using Latent Dirichlet Allocation/Latent Semantic Indexing
- **Polarity and importance** of n-grams in the topics

2.1 Community or subnet identification (1)

The prime advantage of **Explicit Semantic Indexing** (2) during sub-net creation is that it provides greater flexibility. It also does not rely on a specific taxonomy to ingest and categorize documents.

ESI works at a semantic and contextual level rather than a superficial vocabulary of a word or document. It represents the meaning of a piece text, as a combination of the concepts found in the text. The context or meaning of a piece of text, as a composition of the concepts embedded, is best represented by ESI. It has extensive utilities in document classification, semantic relatedness and information retrieval.

In document classification, for example, documents are labeled to make them easier to organize. Labeling a document with keywords/concepts makes it easier to pinpoint. However, keyword labeling alone has its cons; searches carried out using vocabulary with similar semantic context, but different actual terms may not obtain relevant documents. However semantic classification of text reduces the effect on specific terms and can greatly improve the understanding of the text.



Unstructured Data

Explicit Semantic Indexing

Semantic Relatedness Calculation

Subnet 1

Subnet 2

Figure 3.1: Community or Subnet identification (8) process flow

2.1.1 Example for Component 1 (3):

Wikipedia is an extensive source of information where each article can be assumed to be a distinct concept. In ESA based on Wikipedia, a concept is initiated for each article. A vector composed of the terms which prevail in the article then represents the concepts, weighted by their **tf-idf** measure.

The context of any given term can then be projected as a vector of the “association weighting” to the previously established concepts.

By comparing 2 word-vectors using association measures, we can identify their semantic relation to the underlying concepts.

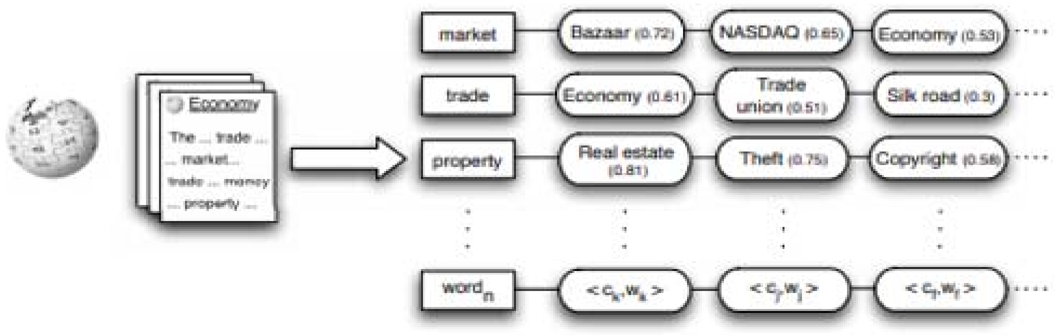
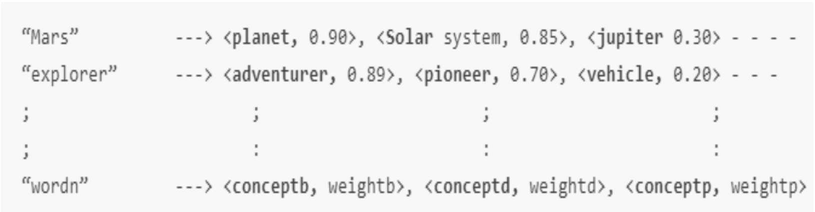


Figure 3.2: Community or Subnet identification example, **Source: Wikipedia**

2.1.2 Representation of Component 1:

Larger documents are projected as a contrast of word vectors obtained from the document vocabulary. Resultant document vectors are called “concept” vectors. For example, a concept vector might look something like the following:



```
“Mars”         ---> <planet, 0.90>, <Solar system, 0.85>, <jupiter 0.30> - - - -
“explorer”     ---> <adventurer, 0.89>, <pioneer, 0.70>, <vehicle, 0.20> - - -
;                    ;                    ;                    ;
;                    :                    :                    :
“wordn”        ---> <conceptb, weightb>, <conceptd, weightd>, <conceptp, weightp>
```

Figure 3.3: Community or Subnet identification representation (1), **Image taken from R software**

Geometrically, a concept vector can be represented as the centroid of the word vectors it is composed of. The image below illustrates the same.

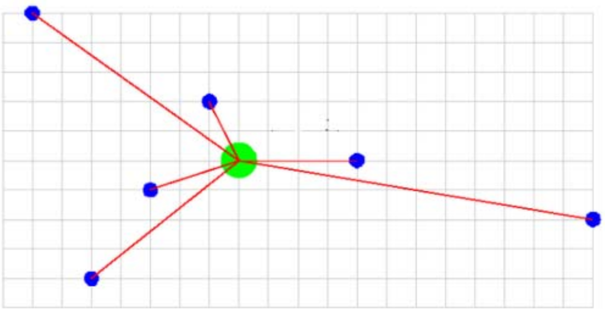


Figure 3.4: Community or Subnet identification representation, Distance of nodes from its centroid, **Image taken from R software.**

Hence, to compare the similarity between two phrases, we can generate their individual concept vectors from their underlying word vectors and then compare these two using association or similarity measures.

2.2 N-Gram Tokenization

In NLP and computational linguistics, an n-gram is an adjoining progression of n items from a fragment of text or speech in question. These items can be either of these – words, letters, syllables, phonemes or any base object according to the requirement.

Sentence: "A swimmer likes swimming thus he swims."

Unigram (1-gram)	Swimmer, likes, swimming, thus, he, swims, …
Bigram (2-gram)	A swimmer, swimmer likes, likes swimming, swimming thus, …
Trigram (3-gram)	A swimmer likes, swimmer likes swimming, likes swimming thus, …

Figure 4: Tokenization example

Most topic model algorithms, like Latent Dirichlet Allocation, are dependent on bag-of-words approach. However, ordering and placement of phrase and words are often useful in understanding the context. Topical n-grams, a topic model that establishes topics as well as topical phrases instead of just bag of words, would be able to capture more context – like the ability to differentiate "white house" in real estate or political context.

The packaging was not good	Unigram	The, packaging, was, not, **good**
	Bigram	The packaging, Packaging was, was not, **not good**

Figure 5.1: How tokenization helps (1)

The product is not expensive	Unigram	The, product, is, not, **expensive**
	Bigram	The product, product is, is not, **not expensive**

Figure 5.2: How tokenization helps (2)

2.2.1 Handling redundancy and noise:

The approach of using n-gram tokenization also has some drawbacks. The huge number of n-grams generated from the raw texts often has few occurrences of n-grams that are extremely rare or occur a lesser number of times. This number is often very high and leads to unnecessary complications and clutter during further model building. Hence a method to select only those n-grams that are semantically sound or have some context helps in solving the above problem.

Detecting tokens is an important problem in the field of computational linguistics, where they're usually called collocations. Most approaches carry out statistical tests that attempt to measure the variance with respect to a random bag-of-words model. **Ted Dunning's G^2** (4) is one such likelihood ratio test which compares the estimated probability of term B occurring after term A to the marginal distribution of term A and B. If the occurrence of A is independent of B, then the probabilities will be almost same. If B is much more frequent after A, then the probabilities will be significantly different.

	Event A	Everything but A
Event B	A and B together (k_11)	B, but not A (k_12)
Everything but B	A without B (k_21)	Neither A nor B (k_22)

Figure 6: Ted Dunning's G^2

Computing the log-likelihood ratio score (also known as G^2) is defined as

LLR = 2 sum(k) (H(k)-H(rowSums(k)) – H(coldSums(k)))

2.3 Weighting the Document Term Matrix

Term Frequency weights are proportional to the occurrence of terms in the document whereas **Inverse document frequency** is an inverse function of the number of documents containing a term thereby quantifying the specificity.

If we are planning to use LDA as the method to observe Topic Models, it must be kept in mind that the weighting cannot be TF-IDF because the basic assumption for Dirichlet distribution demands the support of the Random Variable to be a fractional value and sum of all values should be 1. Hence the only solution is to use Term Frequency approach. But using TF approach can lead to loss of information of rarely used words in the documents, which even though small in number, are significant drivers of conversation. Hence, we plan to use TFIDF at a first level screening to screen words and drop ones that are highly frequent and less specific. Next, with the filtered list of terms, we propose to use its Document-Term Matrix with its Normalized Term Frequency weighting as the input for the Topic Model.

2.4 Topic Modeling

Topic Modeling is an effective method of modeling textual corpora and other sets of discrete data in text format.

The primary aim is to find portrayals of the components of a group that allows dynamic processing of larger sets while sustaining the necessary statistical relationships that can be utilized further for tasks such as text classification, document summarization, relevance judgment prediction etc.

In ML and NLP, a topic model is a type of statistical model which is capable of identifying the hypothetical underlying "topics" that are present in a group of documents thus observing the latent semantic structures in text.

Intuitively, if a given document is related to a particular topic, one can expect a particular set of words from the vocabulary to exist in the given document more frequently.

2.4.1 Latent Semantic Indexing

While the tf-idf measure provides some useful features such as identification of sets of terms/tokens that are discerning for documents, the approach actually does not provide a reasonable reduction in length of its description. It also provides very little information pertaining to intra or inter-document statistical structure.

To address these drawbacks, several dimensionality reduction techniques were proposed, one important such approach being Latent Semantic Indexing (LSI). A singular value decomposition of the document term matrix results in a linear vector space in the tf-idf feature space. This helps to explain a significant amount of variance in the collection of documents.

This approach can be used to obtain a significant reduction in size within large groups. Further, the derived features, which are linear contrasts of the original features, can explain some of the characteristics of synonymy and polysemy (5).

On selecting the top k singular values, and their analogous singular vectors from U and V, we get the approximation to the document term matrix with rank k with a minimal error. Most importantly, now the term and document vectors can be treated as a "semantic space".

We write this approximation as

$$X_k = U_k \Sigma_k V_k^T$$

$$\begin{array}{ccccccccc} & X & & & U & & \Sigma & & V^T \\ & (\mathbf{d}_j) & & & & & & & (\hat{\mathbf{d}}_j) \\ & \downarrow & & & & & & & \downarrow \\ (\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \\ \mathbf{u}_1 \\ \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \\ \mathbf{u}_l \\ \\ \\ \end{bmatrix} \end{bmatrix} & \cdot & \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot & \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix} \end{array}$$

Fig 7: Singular Value Decomposition for LSI, Source: Wikipedia

We can now use the decomposition to:

- Observe how related two documents are by analyzing the vectors by similarity measures
- Compare two terms within a document
- Vector representations of documents and terms can be clustered using different similarity measures.
- Given a query, using it as a reference, and compare it to existing documents in a space of lower dimension.

2.4.2 Advantages of Latent Dirichlet Allocation (6):

Most dimensionality reduction approaches are governed by the fundamental “bag-of-words” assumption where the ordering of words and their occurrence in the text can be neglected.

In the jargon of probability theory, this spells out into the premise of exchangeability for the terms in a document. Although less often formally stated, the assumption of these methods is that the documents are exchangeable, that is, specific ordering of documents in the corpus can be ignored.

De Finetti’s classic representation theorem establishes that any set of random variables which are exchangeable should have a depiction as a mixture distribution - in general – an infinite mixture. Hence, if we take into account exchangeable depictions for terms and documents, it is justified to assume the mixture models that would capture this exchangeability of both terms and documents. This line of thought led to the Latent Dirichlet Allocation (LDA) model

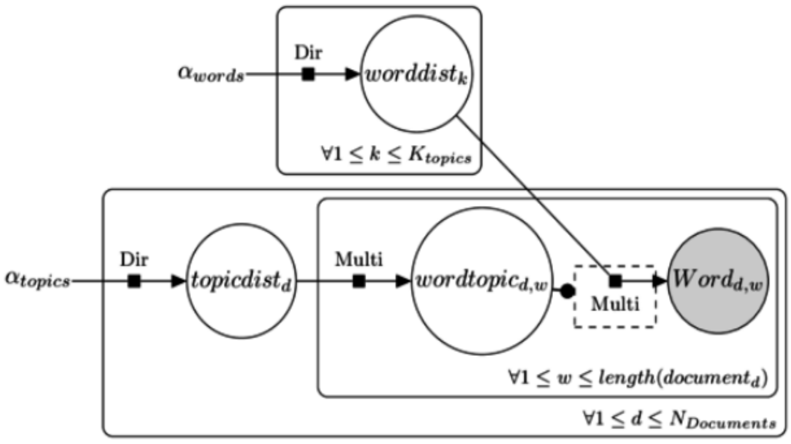

α_{words}
Dir
$worddist_k$
$\forall 1 \leq k \leq K_{topics}$
α_{topics}
Dir
$topicdist_d$
Multi
$wordtopic_{d,w}$
Multi
$Word_{d,w}$
$\forall 1 \leq w \leq length(document_d)$
$\forall 1 \leq d \leq N_{Documents}$

Fig 8: Latent Dirichlet Allocation process flow

The purpose of using LDA/LSI and performing a Topic Model on the set of documents is to have a set of tokens from the top topics which would act as a base for our next step – the driver modeling. Topic modeling would provide us a bag of tokens as the top conversational topics and in the next step, we would be statistically observing the top tokens that are significantly driving the conversation and how it is impacting the ratings.

2.5 Driver Analysis

In general, a key driver analysis is the study of the relations among many factors to identify the primary and significant ones. In our case, we are interested in the relationship between the text tokens which form the documents (here reviews) and the associated ratings that the user provided along with the review/feedback. Such an analysis results, for example, explores and estimates the relationship between the top conversational topics from the review texts and the categorical rating variable.

Now in order to model the tokens from reviews and the categorical variable of ratings, we can use a multinomial logit model where the target/independent variable is ‘Rating’. We can consider either of the following two cases for defining the dependent variable measures:

- TFIDF values of the tokens with respect to the reviews
- Similarity measure (Jaccard, Levenshtein, Cosine) between the tokens and reviews

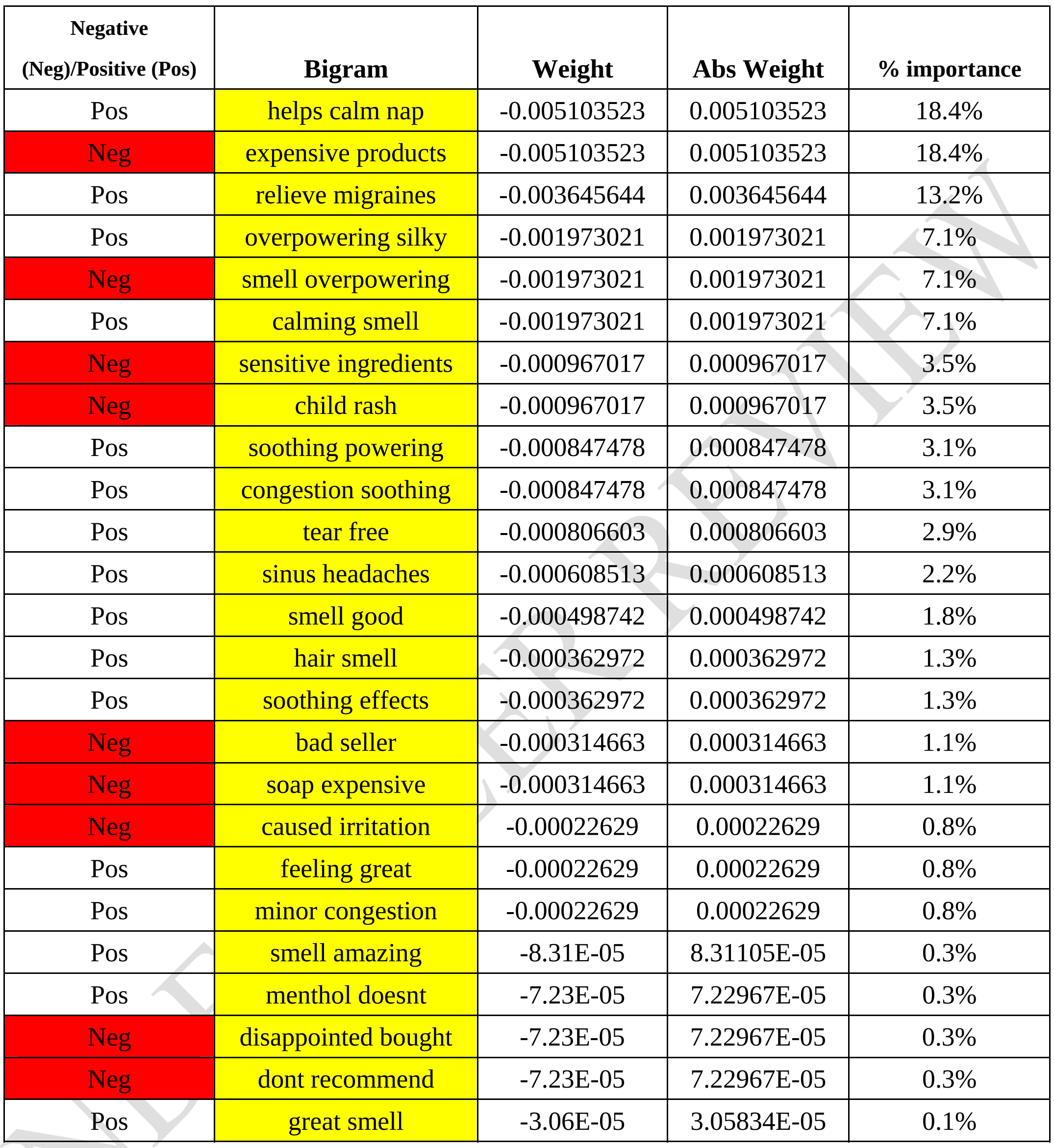
The rationale behind using the Multinomial Logit Model is because of the presence of more than two categories of the independent variable. One more advantage behind this being it would make easier to analyze the drivers between reviews having a very low difference in rating (for example – why a particular set of reviews are rated 5 and another set being rated 4).

The rationale behind using TFIDF values of tokens as dependent variables is that it represents the frequency as well as specificity of the tokens for each document. However, one drawback might be the extreme sparseness of the matrix since the chances of a token occurring in a document (review) decreases when an n-gram of higher order is used. As an alternative to the TFIDF approach, we can use the Similarity approach. This method takes into consideration how a token is semantically similar to each of the review. The setback this method might face is – if the tokens are created without stopwords being removed from the review text and then compared to the entire document, there could be noise and misleading measures of similarity caused by the presence of common words in the reviews. To avoid this over or underestimation of the similarity scores, we can compare the tokens with the pre-processed review texts which contain no stop words.

Once we have the logit model, we can observe the standardized beta estimates of the dependent variables or the tokens in order to have an idea of their contribution in explaining the target or independent variable – ratings (refer fig 9.).

2.6 Sentiment and Mood-State Integration

This component is a further refinement on the drivers observed from the above step. Here we try to classify the drivers into either being a positive or a negative sentiment driver. This would bring deep insights into why a particular token is driving the conversation in a positive/negative direction on the rating scale. Also, if we layer in the notion of mood states into our analysis, then we can have a further idea of how and where a particular driver is affecting the mood state of consumers. The sentiment or mood state integration can be easily implemented using lexicon (7) based word-emotion association.



Negative (Neg)/Positive (Pos)	Bigram	Weight	Abs Weight	% importance
Pos	helps calm nap	-0.005103523	0.005103523	18.4%
Neg	expensive products	-0.005103523	0.005103523	18.4%
Pos	relieve migraines	-0.003645644	0.003645644	13.2%
Pos	overpowering silky	-0.001973021	0.001973021	7.1%
Neg	smell overpowering	-0.001973021	0.001973021	7.1%
Pos	calming smell	-0.001973021	0.001973021	7.1%
Neg	sensitive ingredients	-0.000967017	0.000967017	3.5%
Neg	child rash	-0.000967017	0.000967017	3.5%
Pos	soothing powering	-0.000847478	0.000847478	3.1%
Pos	congestion soothing	-0.000847478	0.000847478	3.1%
Pos	tear free	-0.000806603	0.000806603	2.9%
Pos	sinus headaches	-0.000608513	0.000608513	2.2%
Pos	smell good	-0.000498742	0.000498742	1.8%
Pos	hair smell	-0.000362972	0.000362972	1.3%
Pos	soothing effects	-0.000362972	0.000362972	1.3%
Neg	bad seller	-0.000314663	0.000314663	1.1%
Neg	soap expensive	-0.000314663	0.000314663	1.1%
Neg	caused irritation	-0.00022629	0.00022629	0.8%
Pos	feeling great	-0.00022629	0.00022629	0.8%
Pos	minor congestion	-0.00022629	0.00022629	0.8%
Pos	smell amazing	-8.31E-05	8.31105E-05	0.3%
Pos	menthol doesnt	-7.23E-05	7.22967E-05	0.3%
Neg	disappointed bought	-7.23E-05	7.22967E-05	0.3%
Neg	dont recommend	-7.23E-05	7.22967E-05	0.3%
Pos	great smell	-3.06E-05	3.05834E-05	0.1%

Fig 9: Snapshot of how drivers and associated importance looks after sentiment integration, Screenshot of results from worksheet

3. Conclusion

This research proposed an analytical methodology which enables to understand the drivers of textual data that can be leveraged to drive competitive advantage. The proposed methodology involves various components namely Sub-net Identification, tokenization based on N-grams, calculating the document term matrix using inverse document frequency, topic modeling on each subnet to identify top n-grams and finally polarity & importance of n-grams in the topics – all these components to be performed in a sequential manner resulting in identifying the key drivers along with their polarity (positive or negative) from textual data.

By using the above components in the paper, we have shown the drivers of reviews which impacts the ratings in e-commerce/connected commerce platforms, which inturn helped the business to achieve more traffic on their websites.

As a generalized use case, if there arises a scenario where we have a textual data under study and want to understand the drivers of same, we can implement this approach effectively to attain variable importance of hypernyms (which are drivers of textual data). If there is no target variable per se, we can use the overall sentiment polarity scores to define the target variable of text in question and perform the analysis using the proposed methodology in a similar fashion.

References

1. Blei, D. M., A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. Journal of Machine Learning Research 3: 993–1022
2. Chang, M. W., Ratinov, L., Roth, D., and Srikumar, V. 2008. Importance of semantic representation: dataless classification. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. AAAI Press, Chicago, IL, 830–835.
3. Dumais, S. T. 1994. Latent semantic indexing (lsi) and trec-2. In Proceedings of the Second Text REtrieval Conference (TREC-2). NIST, Gaitherburg, MD, 105–116. Egozi, O., Gabrilovich, E., and Markovitch, S. 2008. Concept-based feature generation and selection for information retrieval. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. AAAI Press, Chicago, IL, 1132–1137.
4. Gabrilovich, E., and Markovitch, S. 2005. Feature generation for text categorization using world knowledge. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05). Morgan Kaufmann Publishers Inc., Edinburgh, Scotland, 1048–1053.
5. Gupta, R., and Ratinov, L. A. 2008. Text categorization with knowledge transfer from heterogeneous data sources. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. AAAI Press, Chicago, IL, 842–847.
6. Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In Proceedings of 14th International Conference on Computational Linguistics.
7. Huang, X., Huang, Y. R., Wen, M., An, A., Liu, Y., and Poon, J. 2006. Applying data mining to pseudo-relevance feedback for high performance text retrieval. In Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM’06). IEEE Computer Society, Hong Kong, 295–306.
8. James, A. C., Zheng, Z. 2014. United States Patent 8832015 B2 Fast binary rule extraction for large scale text data.
9. John, G. H., Kohavi, R., and Pfleger, K. 1994. Irrelevant features and the subset selection problem. In Proceedings of the 11th International Conference on Machine Learning. New Brunswick, NJ, 121–129.
10. Kaptein, R., Kamps, J., and Hiemstra, D. 2008. The impact of positive, negative and topical relevance feedback. In Proceedings of the 17th Text REtrieval Conference (TREC-17). NIST, Gaitherburg, MD.
11. Manfred, S. 2000. The hyperonym problem revisited: Conceptual and lexical hierarchies in language generation. In Association for Computational Linguistics, 93-99.
12. Snow, R., Daniel, J., and Andrew, Y. N. 2004. Learning syntactic patterns for automatic hypernym discovery. In Advances in Neural Information Processing Systems.
13. Ville, B, D. 2013. A Tale of Two SAS Technologies, Generating Maps of Topical Coverage and Linkages. In SAS User Conference Papers Denise Bedford, Kent State University, Cary, NC, 102-112.
14. Ville, B, D., and Bawa, G. S. 2016. United States Patent 9,317,594. Social community identification for automatic document classification
15. Yan, X., J. Guo, Y. Lan, and X. Cheng. 2013. A biterm topic model for short texts. Proceedings of the 22nd international conference on World Wide Web. ACM 1445– 1456.
16. Zhao, W. X., J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. 2011. Comparing Twitter and Traditional Media Using Topic Models. Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 338–349.