Topic models for Sentiment analysis: A Literature Survey

Nikhilkumar Jadhav
123050033

June 26, 2014

In this report, we present the work done so far in the field of sentiment analysis using topic models. The roadmap of the report is as follows. In section 1, we introduce topic models. Section 2 discusses the applications of topic models. Finally, in section 3 we explain how some works have made use of topic models for sentiment analysis.

1 Introduction to Topic models

Bayesian networks are a formal graphical language to express the joint distribution of a system or phenomenon in terms of random variables and their conditional dependencies in a directed graph [Heinrich, 2005]. A generative model is a bayesian network which provides an intuitive description of an observed phenomenon which states how observations could have been generated by realizations of random variables and their propagation along the directed edges of the network [Heinrich, 2005]. Topic models are probabilistic generative models. Topic models try to model the text and usually the generation process of text by means of a bayesian network.

Latent Dirichlet Allocation (*LDA*) is one of the most popular topic models [Blei, Ng, and Jordan, 2003]. Here, the word *latent* signifies capturing the meaning of the text by finding out the hidden topics. It identifies the topic structure in text using co-occurrence structure of terms. It is completely unsupervised as it does not require any background knowledge. The model is as show in figure 1. The replication of a node is represented by a *plate*. This is to account for multiple values or mixture components. Let us have a look at the generative process of *LDA*.

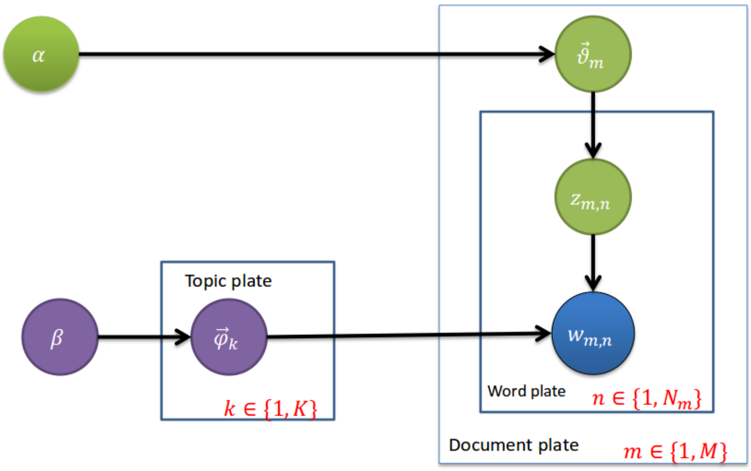

Figure 1: Latent Dirichlet Allocation

1.1 Generative Model for LDA

Let us describe the quantities used in the model

M number of documents to generate (const scalar).
K number of topics/mixture components (const scalar).
V number of terms t in vocabulary (const scalar).
$\vec{\alpha}$ hyper-parameter on the mixing proportions ($K - vector$ or scalar if symmetric).
$\vec{\beta}$ hyper-parameter on the mixing components ($K - vector$ or scalar if symmetric).
$\vec{\vartheta}_m$ parameter notation for p(z|d=m), the topic mixture proportion for document m.
One proportion for each document, $\underline{\theta} = \{\vec{\vartheta}_m\}$ $m = 1 \cdots M$ ($M \times K$ $matrix$).
$\vec{\psi}_k$ parameter notation for p(t|z=k), the mixture component of topic k.
One component for each topic, $\underline{\phi} = \{\vec{\psi}_k\}$ $k = 1 \cdots K$ ($K \times V$ $matrix$).
N_m document length (document-specific), here modeled with a Poisson distribution with constant parameter xi.
$z_{m,n}$ mixture indicator that chooses the topic for the nth word in document m.
$w_{m,n}$ term indicator for the nth word in document m.

1.2 Generative Process for LDA

The generative process for *LDA* is as follows:

```
□ Topic Plate :
for all topics $k \in [1, K]$ do
  sample mixture components $\vec{\psi}_k \sim Dir(\vec{\beta})$
end for
□ Document Plate :
for all documents $m \in [1, M]$ do
  sample mixture proportion $\vec{\vartheta}_m \sim Dir(\vec{\alpha})$
  sample document length $N_m \sim Poiss(\xi)$
  □ Word Plate :
  for all words $n \in [1, N_m]$ in document m do
    sample topic index $z_{m,n} \sim Mult(\vec{\vartheta}_m)$
    sample term for word $w_{m,n} \sim Mult(\vec{\psi}_{z_{m,n}})$
  end for
end for
```

1.3 Generative Process in Simple Terms

- When writing each document, you decide on the number of words N the document will have.
- Choose a topic mixture for the document using dirichlet hypergenerator.
- Generate each word in the document by:
    1. First pick a topic using the multinomial distribution generated from the dirichlet hypergenerator.
    2. Then using the topic and the word-topic distribution to generate the word itself.

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

1.4 Inference via Gibbs Sampling

The exact inference is intractable in case of *LDA*. An approximate inference via *Gibbs* sampling is used. *Gibbs* Sampling [Walsh, 2004] is a special case of *Markov-chain Monte Carlo*. *MCMC* can emulate high dimensional probability distributions, $p(\vec{x})$ by the stationary distribution of a

Markov chain. Each sample is generated for each transition in the chain. This is done after a stationary state of the chain has been reached which happens after a so-called "burn-in period" which eliminates the effect of initialization parameters. In *Gibbs* sampling, the dimensions x_i of the distribution are sampled alternately one at a time, conditioned on the values of all other dimensions, denoted by $\vec{x}_{\neg i}$. The full conditional obtained after inference is as given below,

$$\begin{aligned}
p(z_i = k|\vec{z}_{\neg i}, \vec{w}) &= \frac{p(\vec{w}, \vec{z})}{p(\vec{w}, \vec{z}_{\neg i})} && (1)\\
&= \frac{p(\vec{w}|\vec{z})p(\vec{z})}{p(\vec{w}|\vec{z}_{\neg i})p(\vec{z}_{\neg i})} && (2)\\
&\propto \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})}.\frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})} && (3)\\
&\propto \frac{\Gamma(n_k^{(t)} + \beta_t)\Gamma(\sum_{t=1}^{V} n_{k,\neg i}^{(t)} + \beta_t)}{\Gamma(n_{k,\neg i}^{(t)} + \beta_t)\Gamma(\sum_{t=1}^{V} n_k^{(t)} + \beta_t)}.\frac{\Gamma(n_m^{(k)} + \alpha_k)\Gamma(\sum_{k=1}^{K} n_{m,\neg i}^{(k)} + \beta_t)}{\Gamma(n_{m,\neg i}^{(k)} + \beta_t)\Gamma(\sum_{k=1}^{K} n_m^{(k)} + \beta_t)} && (4)\\
&\propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{(t)} + \beta_t}.\frac{n_{m,\neg i}^{(k)} + \alpha_k}{[\sum_{k=1}^{K} n_{m,\neg i}^{(k)} + \alpha_k] - 1} && (5)
\end{aligned}$$

Multinomial Parameters

The multinomial parameter sets, $\underline{\Theta}$ and $\underline{\phi}$ that correspond to the state of the Markov chain, $M = \vec{w}, \vec{z}$ can be obtained as follows

$$p(\vec{\vartheta}_m|M, \vec{\alpha}) = \frac{1}{Z_{\vartheta_m}} \prod_{n=1}^{N_m} p(z_{m,n}|\vec{\vartheta}_m)p(\vec{\vartheta}_m|\vec{\alpha}) = Dir(\vec{\vartheta}_m|\vec{n}_m + \vec{\alpha}) \tag{6}$$

$$p(\vec{\psi}_k|M, \vec{\beta}) = \frac{1}{Z_{\psi_k}} \prod_{k=1}^{K} p(w_i|\vec{\psi}_k)p(\vec{\psi}_k|\vec{\beta}) = Dir(\vec{\psi}_k|\vec{n}_k + \vec{\beta}) \tag{7}$$

Using the expectation of the dirichlet distribution, $Dir(\vec{\alpha}) = \frac{a_i}{\sum_i a_i}$ in equation 6 and equation 7 we get,

$$\psi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t} \tag{8}$$

$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_t}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k} \tag{9}$$

1.5 Gibbs Sampling Algorithm for LDA

```
□ Initialization
zero all count variables, $n_m\hat{}(k), n_m, n_k\hat{}(t), n_k$
for all documents $m \in [1, M]$ do
  for all words $n \in [1, N_m]$ in document $m$ do
    sample topic index $z_{m,n} = k \sim Mult(1/K)$
    increment document-topic count: $n_m\hat{}(k) + 1$
    increment document-topic sum: $n_m + 1$
    increment topic-term count: $n_k\hat{}(t) + 1$
    increment topic-term sum: $n_k + 1$
  end for
end for
□ Gibbs sampling over burn-in period and sampling period
while not finished do
  for all documents $m \in [1, M]$ do
```

```
    for all words $n \in [1, N_m]$ in document $m$ do
      □ for the current assignment of $k$ to a term $t$ for word $w_{m,n}$ :
      decrement counts and sum:$n_m\hat{}k - 1, n_m - 1, n_k\hat{}(t) - 1, n_k - 1$
      □ multinomial sampling according to equation 5 (decrements from the previous step)
      sample topic index $\bar{k} \sim p(z_i|\vec{z}_{\neg\ i}, \vec{w})$
      □ use the new assignment of $z_{m,n}$ to the term $t$ for word $w_{m,n}$ to:
      increment the counts and sum:$n_m\hat{}k + 1, n_m + 1, n_k\hat{}(t) + 1, n_k + 1$
    end for
  end for
  □ check convergence and read out parameters
  if converged ad $L$ sampling iterations since last read out then
    □ the different parameters read outs are averaged
    read out parameter set $\underline{\phi}$ according to equation 8
    read out parameter set $\underline{\theta}$ according to equation 9
  end if
end while
```

1.6 Gibbs Sampling Algorithm in Simple Terms

We should note one important fact that our hidden variable is z i.e., the topic assignment to each word.

- Go through each document and randomly assign each word in the document one of the K topics.
- This random assignment gives you both **topic-document** distribution and **word-topic** distributions but not very good ones.
- To improve them, we compute two things, $p(t|d)$ and $p(w|t)$.
- After that, reassign each word w a new topic, where the topic t is chosen with probability $p(t|d) \times p(w|t)$
- We assume that all the topic assignments except for the current word is question are correct, and then update the assignment of the current word using the model. After a suitable number of iteration, we get proper topic-document and word-topic distributions.

The trained model can then be used to perform inferencing as described next.

1.7 Inferencing for New Documents

Inferencing is the process of finding out the topic distribution in a new document. Suppose the new document is represented by $\bar{m}$, Let us represent a new document by $\vec{w}$. We need to find out the posterior distribution of topics $\vec{z}$ given the word vector of the document $\vec{w}$ and the *LDA Markov* state, $M = \{\vec{z}, \vec{w}\}$: $p(\vec{z}, \vec{w}; M)$. The algorithm is a modification the *Gibbs* sampling algorithm we saw. It starts of by randomly assigning topics to words and then performs number of loops through the *Gibbs* sampling update (locally for the words i of $\bar{m}$) [Heinrich, 2005].

$$p(\bar{z}_i = k|\bar{w}_i = t, \bar{\vec{z}}_{\neg i}, \bar{\vec{w}}_{\neg i}; M) = \frac{n_k^{(t)} + \bar{n}_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \bar{n}_{k,\neg i}^{(t)} + \beta_t} . \frac{n_{\bar{m},\neg i}^{(k)} + \alpha_k}{[\sum_{k=1}^{K} n_{\bar{m},\neg i}^{(k)} + \alpha_k] - 1} \tag{10}$$

where $\bar{n}_k^t$ counts the observations of term t and topic k in the new document.

The topic distribution of the new document can be found out using the following equation,

$$\vartheta_{\bar{m},k} = \frac{n_{\bar{m}}^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_{\bar{m}}^{(k)} + \alpha_k} \tag{11}$$

Now that we have a basic idea about what topic models are graphically and mathematically, we can discuss some of their applications.

2 Applications of Topic models

LDA gives two outputs, **word-topic** distribution and **topic-document**. The applications mainly use these two outputs. First we will focus on the word-topic distribution

2.1 Applications of Word-Topic Distribution

1. *List of top words in a given topic*
   Using the the word-topic distribution, we can get a list of top n words in any given topic.

2. *Clustering of Words by topic*
   This distribution can also be used to cluster new words based on the co-occurrence principle.

2.2 Applications of Topic-Document Distribution

Let us have a look at the applications which use the topic-document distribution.

1. **Similarity Ranking**
   We can find the topic distribution for each of the document and compare them for similarity. As these are probability distributions, we make use of a modified KL-divergence method to do this.

2. **Querying**
   Querying makes use of similarity ranking to find the documents which are most similar to a given a query.

3. **Clustering**
   Documents can be clustered as per their major topic.

4. **Document classification**
   The topic having highest proportion in the document will be it's class.

In the next section, we will look at some attempts to combine topic and sentiment in a generative modeling framework.

3 Topic models for Sentiment Analysis

We will have a look at the following models one by one.

1. A Generative model for sentiment [Eguchi and Lavrenko, 2006]

2. Topic Sentiment Mixture model [Mei, Ling, Wondra, Su, and Zhai, 2007]

3. Joint Sentiment Topic model [Lin and He, 2009]

4. Aspect-Sentiment Unification model [Jo and Oh, 2011]

3.1 A Generative model for sentiment

Sentiment Analysis has been used in Information Retrieval to improve the performance. Information Retrieval was mainly concerned with factual/objective data. So, intuitively we see that subjectivity classification can aid information retrieval. [Riloff, Wiebe, and Phillips, 2005] has work based on it in which they try to exploit subjectivity analysis to improve performance of information extraction. Corpus models are useful in fetching documents specific to a certain topic. Sometimes a user might need to fetch documents which have a specific sentiment. One such work on sentiment retrieval using generative models is seen in [Eguchi and Lavrenko, 2006]. In this work, they have assumed that user inputs both query terms as well as indicates the desired sentiment polarity in some way. They have combined sentiment and topic relevance models to retrieve documents which are most relevant to such user requests. This approach is very important for sentiment aware information retrieval. The expression of sentiment in the text is topic dependent. Negative review for a voting event may be expressed using *flawed.* On the other hand negative review of politician may be expressed using *reckless.* Sentiment polarity is topic dependent [Engström, 2004]. The adjective *unpredictable* will have a negative orientation in a car review and it will have a positive orientation in a movie review.

Terminology

The goal of the model is to generate a collection of sentences $s_1, s_2, \ldots, s_n$. Every document is composed of words $w_1, w_2, \ldots, w_n$ drawn from the vocabulary V. A binary variable $b_{ij} \in \{S, T\}$ is used to represent whether a word in position j in sentence i is a topic word or a sentiment word. Let x_i be the polarity for the sentence s_i. x_i is a discrete random variable with three outcomes $\{-1, 0, +1\}$. A statement s_i is represented as a set $\{w_i^s, w_i^t, x_i\}$ where w_i^s are the sentiment bearing words, w_i^t are the topic bearing words and x_i is the sentence polarity. The user query will be represented in a similar fashion $\{q_i^s, q_i^t, q^x\}$. Let p denote a unigram language model. P denotes the set of all possible language models. It is the probability simplex. Similarly, let p_x denote the distribution over three possible polarity values and P_x will be the corresponding ternary probability simplex. The function $\pi : P \times P \times P_x \to [0, 1]$ is a function which assigns a probability $\pi(p_1, p_2, p_x)$ to a pair of language models p_1 and p_2 together with p_x.

Generative model of sentiment

A sentence s_i containing words $w_1, w_2, \ldots, w_j, \ldots, w_m$ is generated in the following way:

1. Draw $p_t,\ p_s and\ p_x$ from $\pi(\cdot, \cdot, \cdot)$.
2. Sample x_i from a polarity distribution $p_x(\cdot)$.
3. For each position $j = 1 \ldots m$:
   - if $b_{ij} = T$: draw w_j from $p_t(\cdot)$;
   - if $b_{ij} = S$: draw w_j from $p_s(\cdot)$

The probability of observing the new statement s_i containing words $w_1, w_2, \ldots, w_j, \ldots, w_m$ is given by:

$$\sum_{p_t, p_s, p_x} \pi(p_t, p_s, p_x) p_x(x_i) \prod_{j=1}^{m} \begin{cases} p_t(w_j) & \text{if b}_{\text{ij}}\text{= T} \\ p_s(w_j) & \text{otherwise} \end{cases} \tag{12}$$

The probability functions are dirichlet smoothed models and $\pi(p_1, p_2, p_x)$ is a non-parametric function.

Each sentence is represented as a bag of words model and the model makes strong independence assumptions. But, due to joint probability distribution used it is able to model cooccurrence.

Retrieval using the model

Suppose we are given a collection of statement C and a query $\{q_i^s, q_i^t, q^x\}$ given by the user. The topic relevance model R_t and the sentiment relevance model R_t are estimated. For each word w in a statement within a collection C, these models are estimated as follows:

$$R_t(w) = \frac{P(q^s, q^t \circ w, q^x)}{P(q^s, q^t, q^x)}, R_s(w) = \frac{P(q^s \circ w, q^t, q^x)}{P(q^s, q^t, q^x)} \tag{13}$$

$q \circ w$ means appending w to the list q. The statements are ranked using a variation of cross-entropy,

$$\alpha \sum_v R_t(v) \log p_t(v) + (1 - \alpha) \sum_v R_s(v) \log p_s(v) \tag{14}$$

The experiments using this approach have shown promising results. This shows that sentiment aware IR can benefit from this technique. As corpus models have been widely used in IR, extending and tuning them for SA aware IR can yield good results.

3.2 TSM

[Mei, Ling, Wondra, Su, and Zhai, 2007] proposed the topic sentiment mixture model. Figure 2 shows the model. The *TSM* model helps to address the following problems.

1. **Learning General Sentiment models**
   Learn a sentiment model for positive opinions and a sentiment model for negative opinions, which are general enough to be used in new unlabeled collections.

2. **Extracting Topic models and Sentiment Coverages**
   Given a collection of Weblog articles and the general sentiment models learned, customize the sentiment models to this collection, extract the topic models, and extract the sentiment coverages.

3. **Modeling Topic Life Cycle and Sentiment Dynamics**
   Model the life cycles of each topic and the dynamics of each sentiment associated with that topic in the given collection.

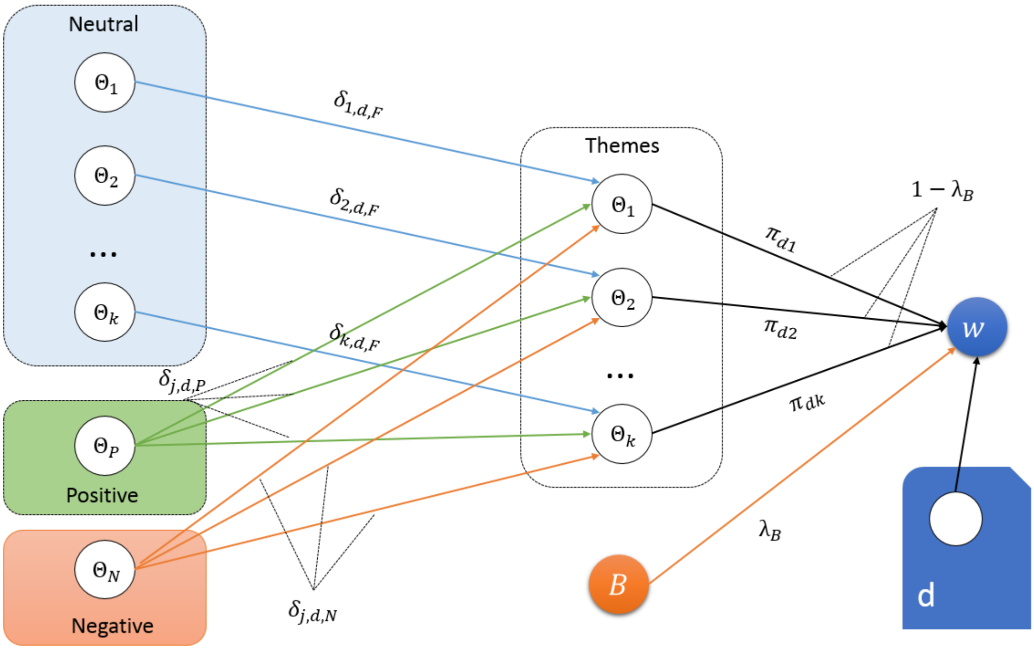

Figure 2: Topic Sentiment Mixture Model

Word Categories

It divides the words in the document into two major classes

1. **Common English words** e.g., “the”, “a”, “of”, etc.

2. **Topic words** eg., “nano”, “price”, “mini” in the document about *iPod*. The topic words are further divided into three classes.

   (a) words about the topic with **neutral** opinion

   (b) words about the topic with **positive** opinion

   (c) words about the topic with **negative** opinion

Multinomials

There are four multinomial distributions to take note of

1. θ_B is a background topic model to capture common English words.
2. $\theta = \theta_1, ..., \theta_k$ are k topic models to capture neutral descriptions about k global subtopics in the collection.
3. θ_P is a positive sentiment model to capture positive opinions for all the topics in the collection.
4. θ_N is a negative sentiment model to capture negative opinions for all the topics in the collection.

Generation Process

An author follows the procedure given below to generate each word of the document.

1. The author would first decide whether the word will be a common English word. If so, the word would be sampled according to θ_B.
2. If not, the author would then decide which of the k subtopics the word should be used to describe.
3. Once the author decides which topic the word is about, the author will further decide whether the word is used to describe the topic neutrally, positively, or negatively.
4. Let the topic picked in step 2 be the jth topic θ_j . The author would finally sample a word using θ_j, θ_P or θ_N, according to the decision in step 3

The accuracy of the model can be increased by providing domain specific lexicons.

3.3 Joint Sentiment Topic model

[Lin and He, 2009] discusses a joint model of sentiment and topics. Figure 3 shows the model.

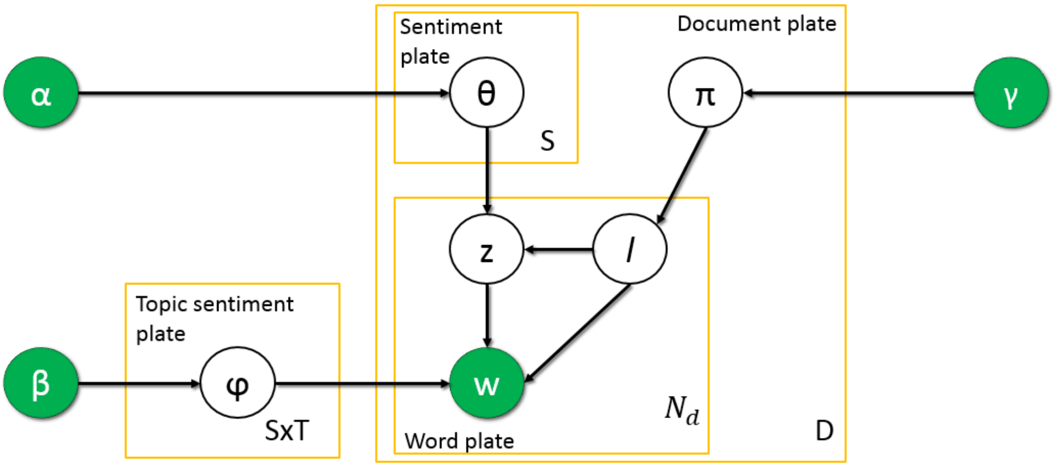

Figure 3: Joint Sentiment Topic model

Assume that we have a collection of D documents denoted by $C = d_1, d_2, \cdots, d_D$; each document in the corpus is a sequence of N_d words denoted by $d = (w_1, w_2, \cdots, w_{N_d})$ and each word in the document is an item from a vocabulary index with V distinct terms denoted by $1, 2, \cdots, V$. Let S and T be the number of distinct sentiment and topic labels respectively. The procedure of generating a document is described as follows.

```
For each document $d$, choose a distribution $\pi_d \sim Dir(\gamma)$.
For each sentiment label $l$ under document $d$, choose a distribution
$\theta_{d,k} \sim Dir(\alpha)$.
For each word $w_i$ in document $d$
  - choose a sentiment label $l_i \sim \pi_d$,
  - choose a topic $z_i \sim \theta_{d,l_i}$,
  - choose a word $w_i$ from the distribution over words defined by the
    topic $z_i$ and sentiment label $l_i$, $\psi_{z_i}\hat{}l_i$
```

The hyper-parameter α in JST is the prior observation count for the number of times topic j is associated with sentiment label l sampled from a document.

The hyper-parameter β is the prior observation count for the number of times words sampled from topic j are associated with sentiment label l.

Similarly, the hyper-parameter γ is the prior observation count for the number of times sentiment label l is associated with a document.

The latent variables of interest in JST are

1. The joint sentiment/topic-document distribution, θ
2. The joint sentiment/topic-word distribution, ϕ
3. The joint sentiment-document distribution, π

To obtain the distributions for θ, ϕ, and π, we firstly estimate the posterior distribution over z i.e, the assignment of word tokens to topics and sentiment labels.

We need to estimate the distribution, $P(z_t = j, l_t = k | w, z_{\neg t}, l_{\neg t}, \alpha, \beta, \gamma)$ where $z_{\neg t}$ and $l_{\neg t}$ are vector of assignments of topics and labels for all words in the collection except for the word position t in document d.

The joint distribution can be given as follows,

$$P(w, z, l) = P(w|z, l)P(z|l) = P(w|z, l)P(z|l, d)P(l|d) \tag{15}$$

After calculations similar to LDA, we get the following full conditional,

$$P(z_t = j, l_t = k | w, z_{\neg t}, l_{\neg t}, \alpha, \beta, \gamma) = \frac{\{N_{i,j,k}\}_{\neg t} + \beta}{\{N_{j,k}\}_{\neg t} + V\beta} . \frac{\{N_{j,k,d}\}_{\neg t} + \alpha}{\{N_{k,d}\}_{\neg t} + T\alpha} . \frac{\{N_{k,d}\}_{\neg t} + \gamma}{\{N_d\}_{\neg t} + S\gamma} \tag{16}$$

where,
V is the size of the vocabulary
T is the number of topics
S is the total number of sentiment labels
D is the number of documents in the collection
$N_{i,j,k}$ is the number of times word i appeared in topic j and with sentiment label k
$N_{j,k}$ is the number of times words are assigned to topic j and sentiment label k
$N_{j,k,d}$ is the number of times a word from document d has been associated with topic j and sentiment label k
$N_{k,d}$ is the number of times sentiment label k has been assigned to some word tokens in document d
N_d is the total number of words in the collection

θ, ϕ, and π can be estimated as follows

$$\phi_{i,j,k} = \frac{N_{i,j,k} + \beta}{N_{j,k} + V\beta} \tag{17}$$

$$\theta_{j,k,d} = \frac{N_{j,k,d} + \alpha}{N_{k,d} + T\alpha} \tag{18}$$

$$\pi_{k,d} = \frac{N_{k,d} + \gamma}{N_d + S\gamma} \tag{19}$$

The Gibbs sampling procedure in this case is similar to that of *LDA*.

JST can be used for document level sentiment classification and topic detection simultaneously. *Joint sentiment topic modeling* is completely unsupervised as compared to existing approaches for sentiment classification. The performance of *JST* on movie review classification is competitive compared to other supervised approaches.

3.4 Aspect Sentiment Unification model

[Jo and Oh, 2011] proposed the aspect-sentiment unification model. It is similar to *JST* model. *JST* does not limit individual words. *JST* is different from *ASUM* in that individual words may come from different language models. In contrast, *ASUM* constrains the words in a single sentence to come from the same language model, so that each of the inferred language models is more focused on the regional co-occurrences of the words in a document. Both *JST* and *ASUM* make use of a small seed set of sentiment words, but the exploitation is not explicitly modeled in *JST*. *ASUM* integrates the seed words into the generative process, and this provides *ASUM* with a more stable statistical foundation

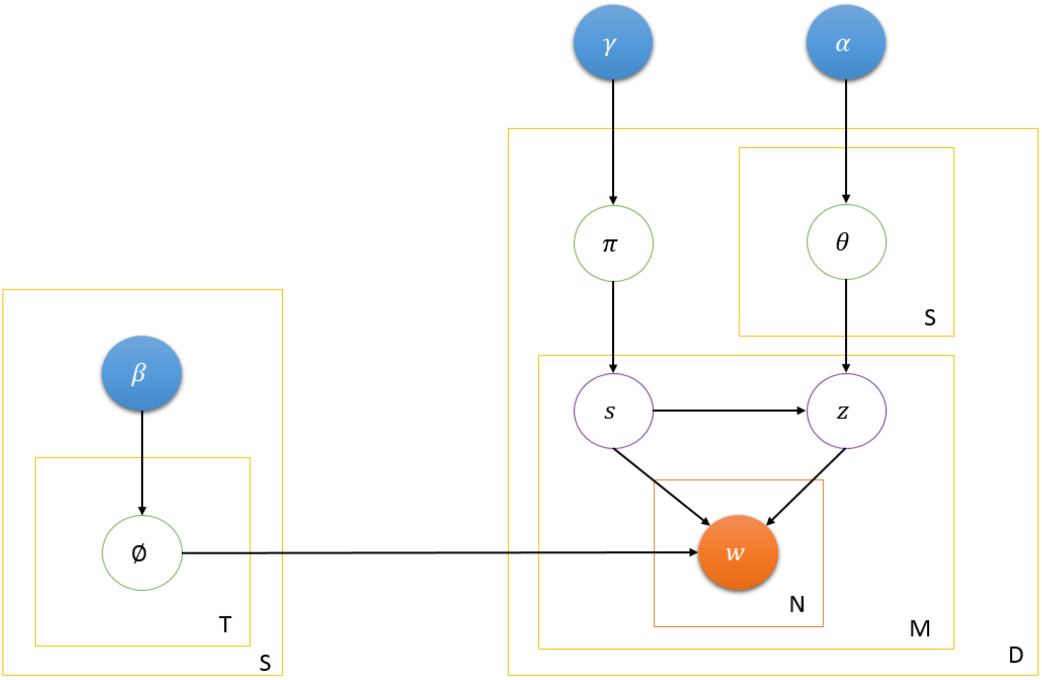

Figure 4: Aspect Sentiment Unification model

SUMMARY

We introduced topic models with *LDA* in section 1. The generative process and the inference method was explained in detail. The various applications of topic models were discussed in section 2. Finally, in section 3 we had a look at the generative models which combine topics and sentiment.

References

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 345–354. Association for Computational Linguistics, 2006.

C. Engström. Topic dependence in sentiment classification. *Unpublished MPhil Dissertation. University of Cambridge*, 2004.

G. Heinrich. Parameter estimation for text analysis. *Web: http://www.arbylon.net/publications/text-est.pdf*, 2005.

Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.

C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.

E. Riloff, J. Wiebe, and W. Phillips. Exploiting subjectivity classification to improve information extraction. In *Proceedings of the National Conference On Artificial Intelligence*, volume 20, page 1106. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

B. Walsh. Markov chain monte carlo and gibbs sampling. 2004.