CONSTRUCTING ASPECT-BASED SENTIMENT LEXICONS WITH TOPIC MODELING

Elena Tutubalina[1] and Sergey I. Nikolenko[1,2,3,4]

[1] Kazan (Volga Region) Federal University, Kazan, Russia

[2] Steklov Institute of Mathematics at St. Petersburg

[3] National Research University Higher School of Economics, St. Petersburg

[4] Deloitte Analytics Institute, Moscow, Russia

April 7, 2016

INTRO: TOPIC MODELING AND SENTIMENT ANALYSIS

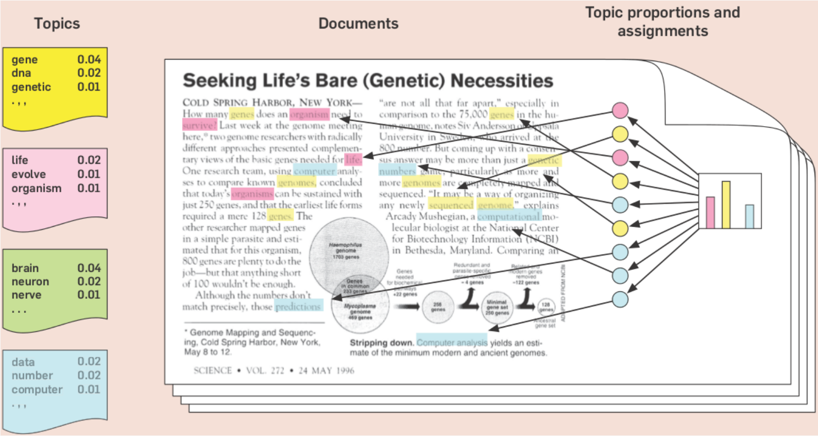
OVERVIEW

- Very brief overview of the paper:
  - we would like to do sentiment analysis;
  - there are topic model extensions that deal with sentiment;
  - but they always rely on an external dictionary of sentiment words;
  - in this work, we show a way to extend this dictionary automatically from that same topic model.

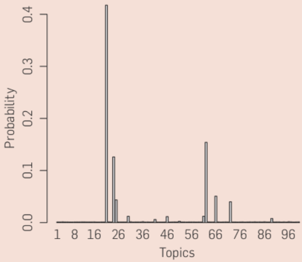
- Sentiment analysis / opinion mining:
  - traditional approaches set positive/negative labels by hand;
  - recently, machine learning models are trained to assign sentiment scores for most words in the corpora;
  - however, they can't really work totally unsupervised, and high-quality manual annotation is expensive;
  - moreover, there are different *aspects*.
- *Problem*: automatically mine sentiment lexicons for specific aspects.

- Latent Dirichlet Allocation (LDA) – *topic modeling* for a corpus of texts:
    - a document is represented as a mixture of topics;
    - a topic is a distribution over words;
    - to generate a document, for each word we sample a topic and then sample a word from that topic;
    - by learning these distributions, we learn what topics appear in a dataset and in which documents.

- Sample LDA result from (Blei, 2012):



- Sample LDA result from (Blei, 2012):

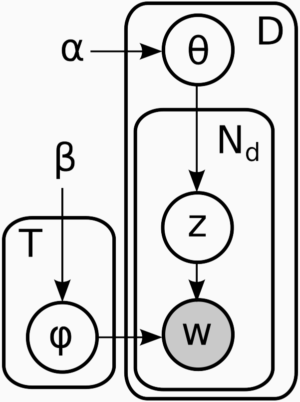


"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

- There are two major approaches to inference in probabilistic models with a loopy factor graph like LDA:
    - *variational approximations* simplify the graph by approximating the underlying distribution with a simpler one, but with new parameters that are subject to optimization;
    - *Gibbs sampling* approaches the underlying distribution by sampling a subset of variables conditional on fixed values of all other variables.
- Both approaches have been applied to LDA.
- We will extend the Gibbs sampling.

- The total likelihood of the LDA model is

$$p(z, w, \alpha, \beta) = \int_{\theta\,,\,\varphi} p(\theta \mid \alpha) p(z \mid \theta) p(w \mid z, \varphi) p(\varphi \mid \beta) d\theta d\varphi.$$



- And in collapsed Gibbs sampling, we sample

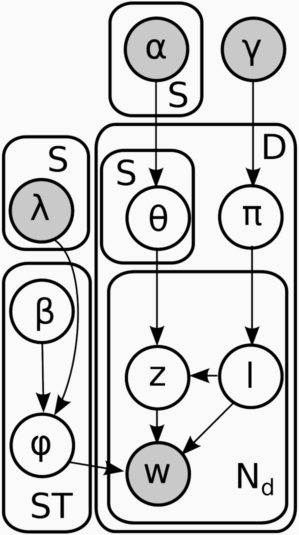
$$p(z_j = t \mid z_{-j}, w, \alpha, \beta) \propto \frac{n_{*,t,d}^{\neg j} + \alpha}{n_{*,*,d}^{\neg j} + T\alpha} \cdot \frac{n_{w,t,*}^{\neg j} + \alpha}{n_{*,t,*}^{\neg j} + W\beta},$$

  where z_{-j} denotes the set of all z values except z_j.
- Samples are then used to estimate model variables:

$$\theta_{td} = \frac{n_{w,t,d} + \alpha}{n_{w,*,d} + T\alpha}, \quad \varphi_{wt} = \frac{n_{w,t,*} + \beta}{n_{*,t,*} + W\beta}.$$

- There exist many LDA extensions:
    - DiscLDA: LDA for classification with a class-dependent transformation in the topic mixtures;
    - Supervised LDA: documents with a response variable, we mine topics that are indicative of the response;
    - TagLDA: words have tags that mark context or linguistic features;
    - Tag-LDA: documents have topical tags, the goal is to recommend new tags to documents;
    - Topics over Time: topics change their proportions with time;
    - hierarchical modifications with nested topics are also important.
- In particular, there are extensions tailored for sentiment analysis.

- JST: topics depend on sentiments from a document's sentiment distribution π_d, words are conditional on sentiment-topic pairs.
- Generative process – for each word position j:
  (1) sample a sentiment label $l_j \sim \mathrm{Mult}(\pi_d)$;
  (2) sample a topic $z_j \sim \mathrm{Mult}(\theta_{d,l_j})$;
  (3) sample a word $w \sim \mathrm{Mult}(\varphi_{l_j,z_j})$.

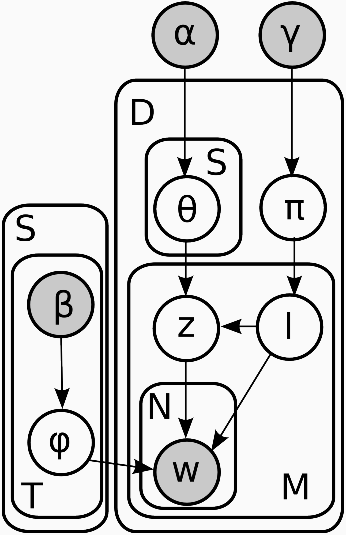


- In Gibbs sampling, one can marginalize out π_d:

$$p(z_j = t, l_j = k \mid z_{-j}, w, \alpha, \beta, \gamma, \lambda) \propto \frac{n^{\neg j}_{*,k,t,d} + \alpha_{tk}}{n^{\neg j}_{*,k,*,d} + \sum_t \alpha_{tk}} \cdot \frac{n^{\neg j}_{w,k,t,*} + \beta_{kw}}{n^{\neg j}_{*,k,t,*} + \sum_w \beta_{kw}} \cdot \frac{n^{\neg j}_{*,k,*,d} + \gamma}{n^{\neg j}_{*,*,*,d} + S\gamma},$$

where $n_{w,k,t,d}$ is the number of words w generated with topic t and sentiment label k in document d, α_{tk} is the Dirichlet prior for topic t with sentiment label k.

- ASUM: aspect-based analysis + sentiment for user reviews; a review is broken down into sentences, assuming that each sentence speaks about only one aspect.
- Basic model – Sentence LDA (SLDA): for each review d with topic distribution θ_d, for each sentence in d,
  (1) choose its sentiment label $l_s \sim \mathrm{Mult}(\pi_d)$,
  (2) choose topic $t_s \sim \mathrm{Mult}(\theta_{dl_s})$ conditional on the sentiment label l_s,
  (3) generate words $w \sim \mathrm{Mult}(\varphi_{l_s t_s})$.



- Denoting by $s_{k,t,d}$ the number of sentences (rather than words) assigned with topic t and sentiment label t in document d:

$$
\begin{aligned}
p(z_j = t, l_j = k \mid l_{-j}, z_{-j}, w, \gamma, \alpha, \beta) \propto \\
\frac{s_{k,t,d}^{\neg j} + \alpha_t}{s_{k,*,d}^{\neg j} + \sum_t \alpha_t} \cdot \frac{s_{k,*,d}^{\neg j} + \gamma_k}{s_{*,*,d}^{\neg j} + \sum_{k'} \gamma_{k'}} \times \\
\times \frac{\Gamma\left(n_{*,k,t,*}^{\neg j} + \sum_w \beta_{kw}\right)}{\Gamma\left(n_{*,k,t,*}^{\neg j} + \sum_w \beta_{kw} + W_{*,j}\right)} \prod_w \frac{\Gamma\left(n_{w,k,t,*}^{\neg j} + \beta_{kw} + W_{w,j}\right)}{\Gamma\left(n_{w,k,t,*}^{\neg j} + \beta_{kw}\right)},
\end{aligned}
$$

  where $W_{w,j}$ is the number of words w in sentence j.
- There are other models and extensions (USTM).

LEARNING SENTIMENT PRIORS

- All of the models above assume that we have prior sentiment information from an external vocabulary:
  - in JST and Reverse-JST, word-sentiment priors λ are drawn from an external dictionary and incorporated into β priors; $\beta_{kw} = \beta$ if word w can have sentiment label k and $\beta_{kw} = 0$ otherwise;
  - in ASUM, prior sentiment information is also encoded in the β prior, making β_{kw} asymmetric similar to JST;
  - the same holds for other extensions such as USTM.

IDEA

- Dictionaries of sentiment words do exist.
- But they are often incomplete; for instance, we wanted to apply it to Russian where there are few such dictionaries.
- It would be great to extend topic models for sentiment analysis to train sentiment for new words automatically!
- We can assume access to a small seed vocabulary with predefined sentiment, but the goal is to extend it to new words and learn their sentiment from the model.

IDEA

- In all of these models, word sentiments are input as different β priors for sentiment labels.
- If only we could train these priors automatically...

IDEA

- In all of these models, word sentiments are input as different β priors for sentiment labels.
- If only we could train these priors automatically...
- ...and we can do it with EM!

GeneralEMScheme

```
while inference has not converged do
    for N steps do                                    □ M-step
        run one Gibbs sampling update step
    update β_kw priors                                □ E-step
```

- This scheme works for every LDA extension considered above.
- At the E-step, we update $\beta_{kw} \propto n_{w,k,*,*}$, and we can choose the normalization coefficient ourselves, so we start with high variance and then gradually refine β_{kw} estimates in simulated annealing:

$$\beta_{kw} = \frac{1}{\tau} n_{w,k,*,*},$$

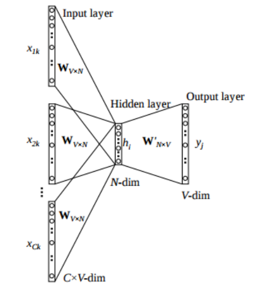
where τ is a regularization coefficient (temperature) that starts large (high variance) and then decrease (lower variance).

EM TO TRAIN β

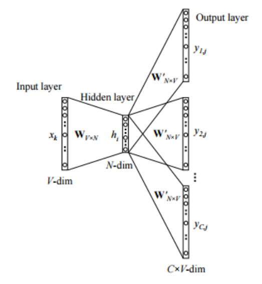
- Thus, the final algorithm is as follows:
  - start with some initial approximation to β_s^w (from a small seed dictionary and maybe some simpler learning method used for initialization and then smoothed);
  - then, iteratively,
    - at the E-step of iteration i, update β_{kw} as $\beta_{kw} = \frac{1}{\tau(i)} n_{w,k,*,*}$ with, e.g., $\tau(i) = max(1, 200/i)$;
    - at the M-step, perform several iterations of Gibbs sampling for the corresponding model with fixed values of β_{kw}.

- Earlier (MICAI 2015), we have shown that this approach leads to improved results in terms of sentiment prediction quality.
- In this work, we use improved sentiment-topic models to learn new aspect-based sentiment dictionaries.
- To do so, we used *distributed word representations* (word embeddings).

- Distributed word representations map each word occurring in the dictionary to a Euclidean space, attempting to capture semantic relationships between the words as geometric relationships in the Euclidean space.
- Started back in (Bengio et al., 2003), exploded after the works of Bengio et al. and Mikolov et al. (2009–2011), now used everywhere; we use embeddings trained on a very large Russian dataset (thanks to Nikolay Arefyev and Alexander Panchenko!).



CBOW



skip-gram

- Intuition: words similar in some aspects of their meaning, e.g., sentiment, will be expected to be close in the semantic Euclidean space.
- To expand the top words of resulting topics:
  - extract word vectors for all top words from the distribution φ in topics and all words in available general-purpose sentiment lexicons;
  - for every top word in the topics, construct a list of its nearest neighbors according to the cosine similarity measure in the $\mathbb{R}^{500}$ space among the sentiment words from the lexicons (20 neighbors is almost always enough).
- We have experimented with other similarity metrics (L_1, L_2, variations on L_∞) with either worse or very similar results.

EXPERIMENTS

DATASET

- Dataset with Russian language reviews on restaurants released for the SentiRuEval-2015 task (Loukachevitch et al., 2015).
- In total, $17{,}132$ unlabeled reviews were used to train the Reverse-JST model.
- Preprocessing natural for topic modeling: remove stopwords and punctuation, convert to lowercase, normalize the text with Mystem, remove too rare words (< 3 occurrences).
- For initial β priors, we used a manually constructed sentiment lexicon.

SAMPLE TOPICS

#	sent.	sentiment words
1	neu	соус [sauce], салат [salad], кусочек [slice], сыр [cheese], тарелка [plate], овощ [vegetable], масло [oil], лук [onions], перец [pepper]
	pos	приятный [pleasant], атмосфера [atmosphere], уютный [cozy], вечер [evening], музыка [music], ужин [dinner], романтический [romantic]
	neg	ресторан [restaurant], официант [waiter], внимание [attention], сервис [service], обращать [to notice], обслуживать [to serve], уровень [level]
2	neu	столик [table], заказывать [to order], вечер [evening], стол [table], приходить [to come], место [place], заранее [in advance], встречать [to meet]
	pos	место [place], хороший [good], вкус [taste], самый [most], приятный [pleasant], вполне [quite], отличный [excellent], интересный [interesting]
	neg	еда [food], вообще [in general], никакой [none], заказывать [to order], оказываться [appear], вкус [taste], ужасный [awful], ничто [nothing]
3	neu	девушка [girl], спрашивать [to ask], вопрос [question], подходить [to come], официантка [waitress], официант [waiter], говорить [to speak]
	pos	большой [big], место [place], выбор [choice], хороший [good], блюдо [dish], цена [price], порция [portion], небольшой [small], плюс [plus]
	neg	цена [price], обслуживание [service], качество [quality], уровень [level], кухня [kitten], средний [average], ценник [price tag], высоко [high]

- The resulting aspect-based lexicons contain 726 topical aspects commonly divided into three types:
  (1) explicit aspects that denote parts of a product (e.g., *сотрудник* [*worker*], *баранина* [*lamb*], *овощ* [*vegetable*], *мексиканский* [*mexican*]);
  (2) implicit aspects that refer indirectly to a product (e.g., *чисто* [*clean*], *ароматный* [*aromatic*], *сытно* [*filling*], *шумно* [*noisy*]);
  (3) narrative words which related to major topics in the text and indirectly refer to sentiment polarity of the text (e.g., *пересолить* [*to oversalt*], *пожелать* [*to wish*], *почувствовать* [*to sense*], *отсутствовать* [*be missing*]).
- Next we applied the mined aspects to sentiment classification to see whether there is an improvement.

- Classifier from (Ivanov, Tutubalina et al., 2015) based on a max-entropy model.
- It uses term frequency features in the context of an aspect term and lexicon-based features.
- Specifically, the following features from an aspect's context window of 4 words:
  (1) lowercased *character n-grams* with document frequency greater than two;
  (2) *lexicon-based unigrams* and *context unigrams and bigrams*;
  (3) *aspect-based bigrams* as a combination of the aspect terms itself and words;
  (4) *lexicon-based features*: the maximal sentiment score, the minimum sentiment score, the total and averaged sums of the words' sentiment scores.

- We compare classifiers with lexicon-based features:
  1. computed on a manually constructed general-purpose lexicon (baseline classifier),
  2. computed on a general-purpose lexicon for all words and aspect-based lexicons for individual aspects.
- We evaluated three different versions of sentiment scores:
  1. *scoresDict*: take sentiment score from the manually created lexicon if the word occurs in the lexicon with a positive or negative label; otherwise, set the score to 0;
  2. *scoresMult*: set the sentiment score of a word as a product of the dictionary score and the similarity;
  3. *scoresCos*: set the sentiment score to cosine similarity score if similarity between the word in question and *хороший* [*good*] is higher than similarity with *плохой* [*bad*]; otherwise, shift sentiment score towards the opposite polarity.

Max-Entropy Classifier	micro-averaged			macro-averaged		
	P	R	F1	P	R	F1
baseline - Lexicon1	0.595	0.344	0.436	0.738	0.649	0.676
scoresDict	0.592	0.344	0.436	0.737	0.649	0.676
scoresMult	0.600	0.351	0.442	0.740	0.653	0.680
scoresCos	**0.610**	0.372	0.462	**0.748**	0.663	**0.691**
baseline - Lexicon2	0.572	0.341	0.427	0.727	0.646	0.671
scoresDict	0.568	0.345	0.430	0.725	0.647	0.672
scoresMult	0.556	0.338	0.420	0.719	0.643	0.667
scoresCos	0.566	0.368	0.447	0.725	0.657	0.680
baseline - Lex1 + Lex2	0.594	0.348	0.439	0.738	0.651	0.679
scoresDict	0.595	0.376	0.461	0.741	0.663	0.689
scoresMult	0.590	0.372	0.457	0.738	0.661	0.687
scoresCos	0.602	**0.376**	**0.463**	0.744	**0.664**	0.690

SAMPLE ASPECT-RELATED SENTIMENT WORDS

aspect	sentiment words
баранина [lamb]	вкусный [tasty], сытный [filling], аппетитный [delicious], душистый [sweet smelling], деликатесный [speciality], сладкий [sweet]
караоке [karaoke]	музыкальный [musical], попсовый [pop], классно [awesome], развлекательный [entertaining], улетный [mind-blowing]
пирог [pie]	вкусный [tasty], аппетитный [delicious],обсыпной [bulk], сытный [filling], черствый [stale], ароматный [aromatic], сладкий [sweet]
ресторан [restaurant]	шикарный [upscale], фешенебельный [fashionable], уютный [cozy], люкс [luxe], роскошный [luxurious], недорогой [affordable]
вывеска [sign]	обветшалый [decayed], выцветший [faded], аляповатый [flashy], фешенебельный [fashionable], фанерный [veneer]
администратор [manager]	люкс [luxe], неисполнительный [careless], ответственный [responsible], компетентный [competent], толстяк [fatty]
интерьер [interior]	уют [comfort], уютный[cozy], стильный [stylish], просторный [spacious], помпезный [magnific], роскошный [luxurious], шикарный [upscale]
вежливый [delicate]	вежливый [delicate], учтивый[polite], обходительный [affable], доброжелательный [good-minded], тактичный [diplomatic]

CONCLUSION

- We have presented a method for automatically extracting aspect-based sentiment lexicons based on an extension of sentiment-related topic models augmented with similarity search based on distributed word representations.
- We extract important new sentiment words for aspect-specific lexicons and show improvements in sentiment classification on standard benchmarks.
- Future work:
    - can we train a more informative relation between sentiment priors and distributed word representations?
    - maybe distributed word representations can be fed directly into the priors?

Thank you for your attention!