Applying Topic Segmentation to Document-Level IR

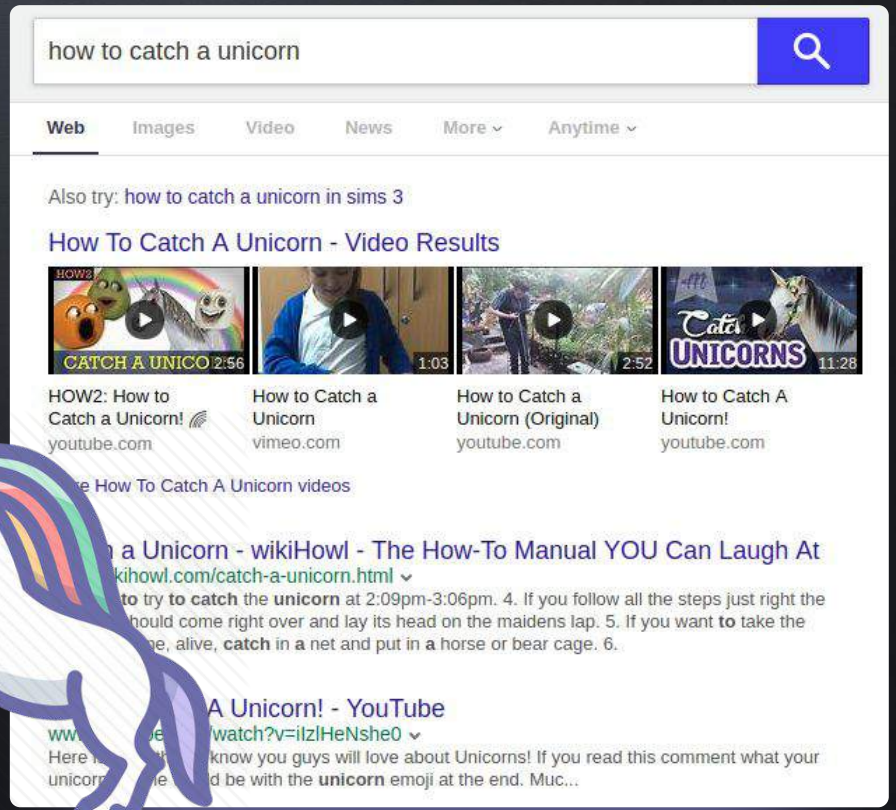
Software Engineering Conference Russia 2018
October 12-13
Moscow

IRELA
Gennady Shtekh
Polina Kazakova
Nikita Nikitinsky

MSU
Nikolay Skachkov

What is IR

Information Retrieval:
matching a query with relevant documents



Our Case

01 /

Document-Level Information Retrieval: query is also a document

02 /

Non-conventionalized term

03 /

Querying by example

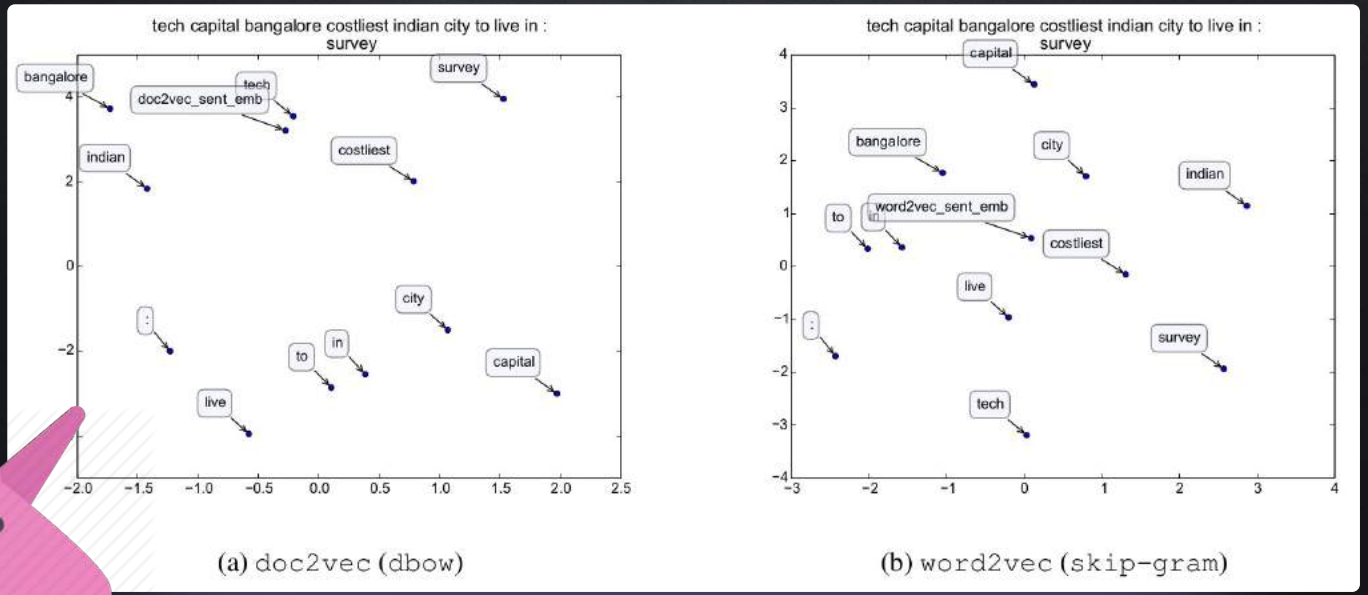
Our Hypothesis

Information Retrieval quality increases by using topic segmentation of documents.

Topic segmentation: splitting texts into semantically homogeneous blocks.

Why to do this? Quality of machine learning algorithms on short (and topically coherent) texts is supposed to be better.

Why Do we Think so?



Example:
document embeddings

Lau, Jey Han, and Timothy Baldwin. "An empirical evaluation of doc2vec with practical insights into document embedding generation." arXiv preprint arXiv:1607.05368 (2016).

SECR

Topic Models

$$p(w \mid d) = \sum_{t \in T} p(w \mid t, d)\, p(t \mid d) = \sum_{t \in T} p(w \mid t)\, p(t \mid d) = \sum_{t \in T} \varphi_{wt} \theta_{td}.$$

01 /

Soft-clustering

02 /

Every document is described as a mixture of topics

03 /

Every topic is described as a mixture of words

04 /

EM-algorithm

05 /

PLSA, LDA

06 /

ARTM - regularizers

Topic Segmentation Pipeline

01 /

Skachkov, N., Vorontsov, K. Improving topic models with segmental structure of texts. [1]

02 /

Based on ARTM Additive Regularization of Topic Models [2] (BigARTM tool [3])

Topic Segmentation Pipeline

Part #1 /

Constructing topic model under sparsity assumption

Gradual estimation of segments borders:

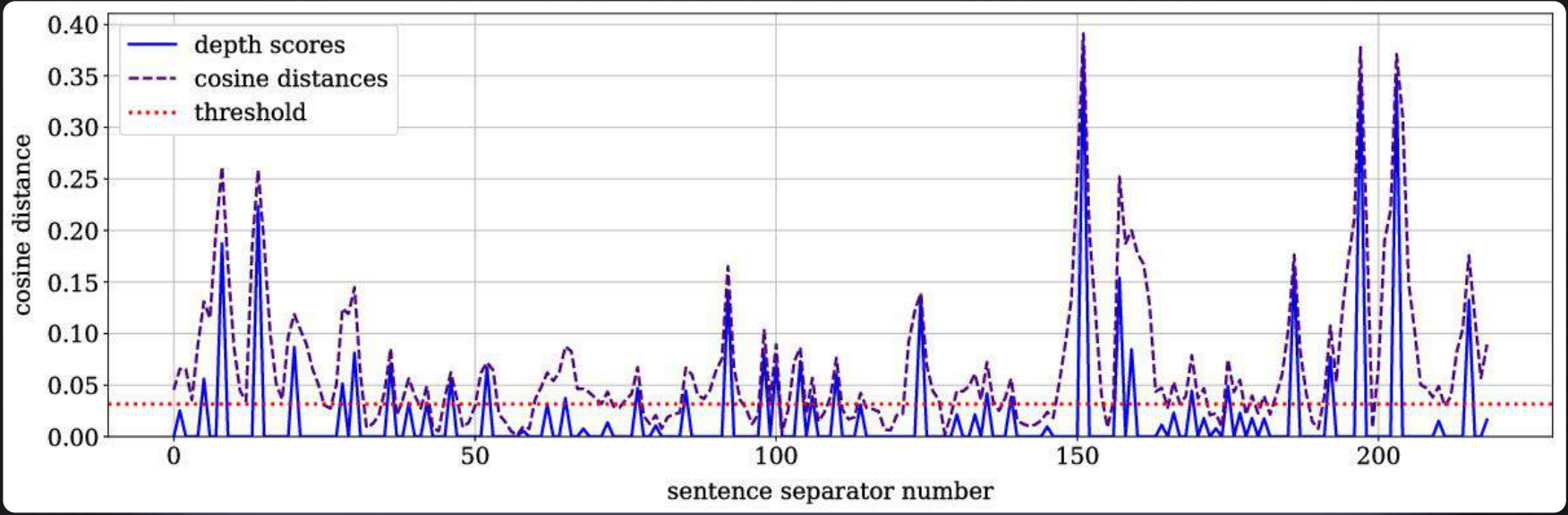
- ★ first, use sentence borders
- ★ then merge adjacent segments if they have the same topics

Topic Segmentation Pipeline

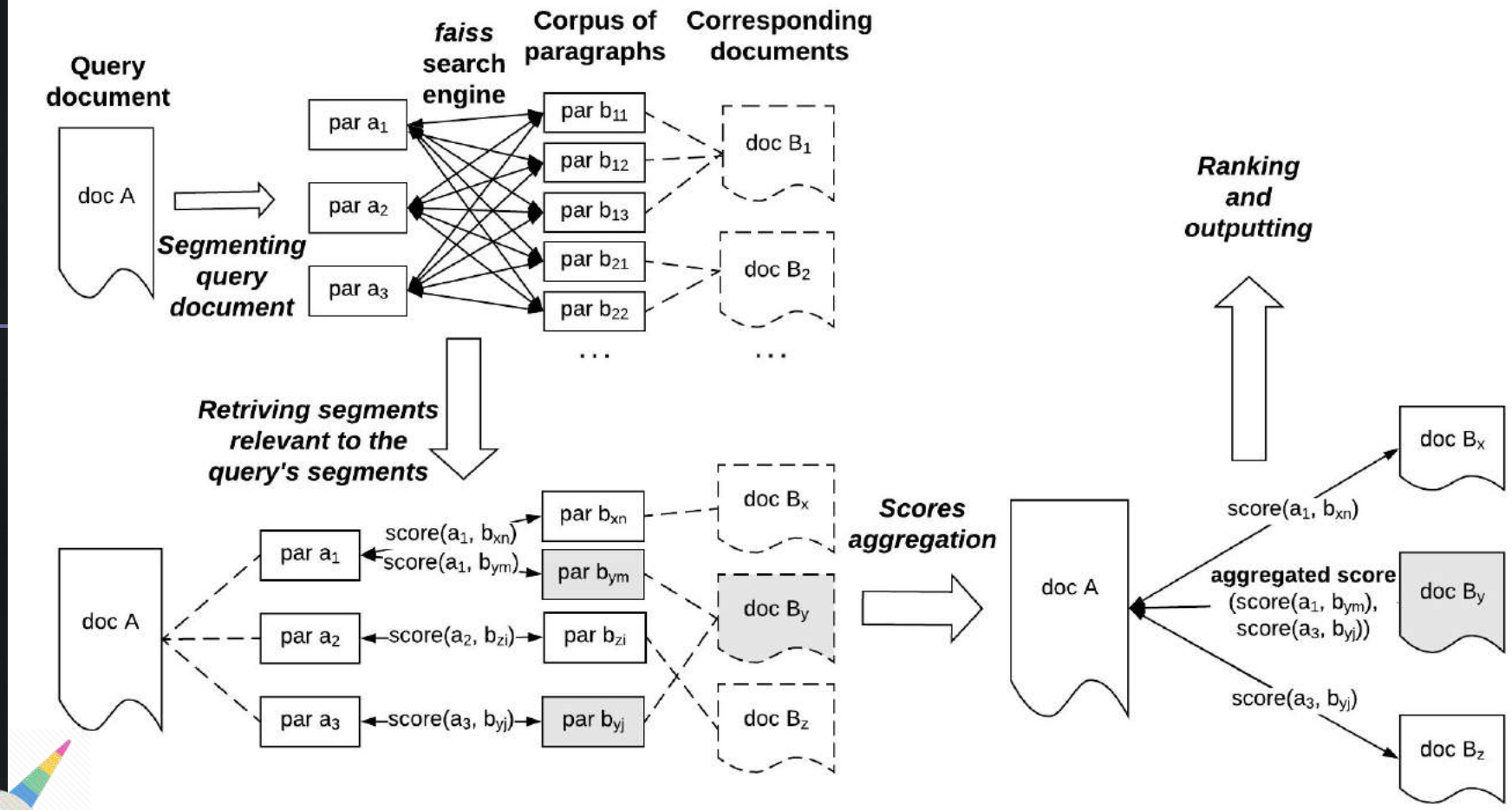
Part #2 / Topic Tiling [4] algorithm:

- ★ For each sentence boundary, consider left and right windows of a length n and compute distance
- ★ Smoothe the distances - depth scores (*ds*)
- ★ Calculate threshold:
  - threshold = mean(*ds*) - *alpha**sqrt(sd(*ds*))
  - *alpha* is varied to change the granularity (default - 0.5)
- ★ Sentence separators with scores more than the threshold - segments boundaries

Topic Segmentation Pipeline



Retrieval Pipeline


Query document
doc A
Segmenting query document
faiss search engine
Corpus of paragraphs
Corresponding documents
par a1
par a2
par a3
par b11
par b12
par b13
par b21
par b22
doc B1
doc B2
Retriving segments relevant to the query's segments
doc A
par a1
par a2
par a3
score(a1, bxn)
score(a1, bym)
score(a2, bzi)
score(a3, byj)
par bxn
par bym
par bzi
par byj
doc Bx
doc By
doc Bz
Scores aggregation
doc A
Ranking and outputting
score(a1, bxn)
aggregated score
(score(a1, bym),
score(a3, byj))
score(a3, byj)
doc Bx
doc By
doc Bz

Experiments: Data

Data
arXiv preprints (140000 preprints: train - 95000, test - 45000)

Preprocessing
spaCy + some manual rules (removing mathematical symbols and short strings)

Test set
triplets: query paper - relevant paper - non-relevant paper; based on arXiv subjects (15715 triplets)

! The same technique was used in Andrew M Dai, Christopher Olah, and Quoc V Le. 2015. Document embedding with paragraph vectors. [6]

Experiments: Baselines

- ARTM segment model (the only model trained on our data!)

Pretrained models:

- averaged word2vec (simple and normalized)
- averaged GloVe (simple and normalized)
- fastText (averaged and original)
- doc2vec
- sent2vec

Experiments: Evaluation

Aggregation

- **Mean:** relevance score = mean of paragraphs scores
- **Best N:** relevance score = mean of N most relevant paragraphs scores (N = 1, 3, 5)

Granularity

- **Two ‘granulites’:** coarser (alpha = 0.3) and finer (alpha = 0.5)

Evaluation

- **Accuracy:** proportion of correctly ranked pairs of documents in total number of triplets

SECR

Results

Model	No segm.	Finer segmentation				Coarser segmentation			
		best 1	best 3	best 5	mean	best 1	best 3	best 5	mean
ARTM	0.817	0.761	0.771	0.773	0.780	0.765	0.772	0.774	0.783
sent2vec	0.770	0.807	0.808	0.807	0.783	0.808	0.809	0.807	0.775
fastText	0.751	0.784	0.785	0.782	0.684	0.784	0.785	0.782	0.680
doc2vec	0.814	0.783	0.785	0.782	0.628	0.780	0.781	0.778	0.636
avW2V	0.817	0.820	0.824	0.822	0.774	0.821	0.823	0.822	0.768
avnormW2V	0.580	0.584	0.583	0.587	0.620	0.584	0.582	0.586	0.616
avGloVe	0.779	0.779	0.779	0.778	0.712	0.777	0.778	0.777	0.709
avnormGloVe	0.573	0.601	0.609	0.609	0.588	0.602	0.609	0.610	0.589
avfasText	0.662	0.746	0.751	0.746	0.638	0.746	0.750	0.745	0.632

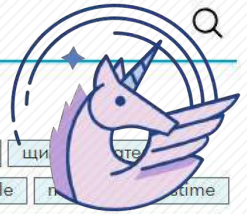
Results and observations

- Segmentation does improve the majority of the models
- ARTM-based model works better on whole texts
- Small values of N in aggregation are probably better
- Influence of segmentation granularity is unclear
- Influence of text style perhaps must be taken into account
- word2vec beats everything (???)

Real Applications: Our Experience

- ★ Cross-lingual search engine
- ★ Ad-hoc retrieval task

как поймать единорога



Контекстное расширение поискового запроса

выследить | убить | убежать | перехитрить | украсть | достать | заманить | затащить | поймал | утащить | дракон | щи… | …ате
восставать | грифон | олень | леопард | червленый | держать | журавль | коронованный | unawares | blacklist | rule | n… | …time
punish | distract | intercept | reason | unsuspecting | dragon | wyvern | cockatrice | basilisk | beast | seahorse | dragonslayer | bestiary
dragonette | minotaur

274 результата (показано **274**)

рус | анг

Единорог

Единоро́г или инро́г (др.-евр. ראם; др.-греч. μονό-κερως; лат. unicornis) — мифическое существо, символизирует целомудрие, в широком смысле духовную чистоту и искания. Чаще всего его представляют в виде коня с одним рогом, выходящим изо лба. == Описание и символика == === У древних авторов === Самым ранним изображениям единорогов больше 4 тысяч лет, найдены в Индии. Затем стали появляться в мифах Западной Азии. В древней Греции и Древнем Риме считались реально существующими животными. Изображения единорога, попадающиеся на древнеегипетских памятниках и на скалах южной Африки, являются ри

66.959% соответствие запросу Tue Jun 26 16:09:15 UTC 2018 Подробнее →

Unicorn

The unicorn is a legendary creature that has been described since antiquity as a beast with a single large, pointed, spiraling horn projecting from its forehead. The unicorn was depicted in ancient seals of the Indus Valley Civilization and was mentioned by the ancient Greeks in accounts of natural history by various writers, including Ctesias, Strabo, Pliny the Younger, and Aelian. The Bible also describes an animal, the re'em, which some versions translate as unicorn.In European folklore, the unicorn is often depicted as a white horse-like or goat-like animal with a long horn and cloven hoov

66.688% соответствие запросу Sun Jul 15 15:53:12 UTC 2018 Подробнее →

References

01 / Nikolay Skachkov and Konstantin Vorontsov. 2018. Improving topic models with segmental structure of texts. In Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue” (2018). 652–661.

02 / Konstantin Vorontsov and Anna Potapenko. 2015. Additive regularization of topic models. Machine Learning 101, 1-3 (2015), 303–323.

03 / Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. 2015. Bigartm: Open source library for regularized multimodal topic modeling of large collections. In International Conference on Analysis of Images, Social Networks and Texts. Springer, 370–381.

04 / Martin Riedl and Chris Biemann. 2012. Text segmentation with topic models. Journal for Language Technology and Computational Linguistics 27, 1 (2012), 47–69.

05 / Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. arXiv preprint arXiv:1702.08734 (2017).

06 / Andrew M Dai, Christopher Olah, and Quoc V Le. 2015. Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998 (2015)

Links

spaCy:	spacy.io/
word2vec:	github.com/jhlau/doc2vec#pre-trained-word2vec-models
GloVe:	nlp.stanford.edu/projects/glove/
fastText:	fasttext.cc/docs/en/english-vectors.html
doc2vec:	github.com/jhlau/doc2vec#pre-trained-doc2vec-models
sent2vec:	github.com/epfml/sent2vec#downloading-pre-trained-models

Contact us

Follow IRELA on:

Telegram: t.me/irelaru
Medium: medium.com/@irela
Facebook: facebook.com/irelaru

Gmail: kazakova1537@gmail.com
Telegram: t.me/brnzz