

# Query Segmentation

Divya Spoorthy  
Siddhartha Mishra  
Raaghav R.

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

# Problem Statement

Query Segmentation is the task of breaking (Segment) the query into multiple adjacent phrases so that each segment can refer to something meaningful, and helps in specifying an important aspect of the query.

- A query segmentation algorithm breaks the input query into, typically, a non-overlapping sequence of words (segments) and those segments are utilized to improve information retrieval performance,

# Challenges

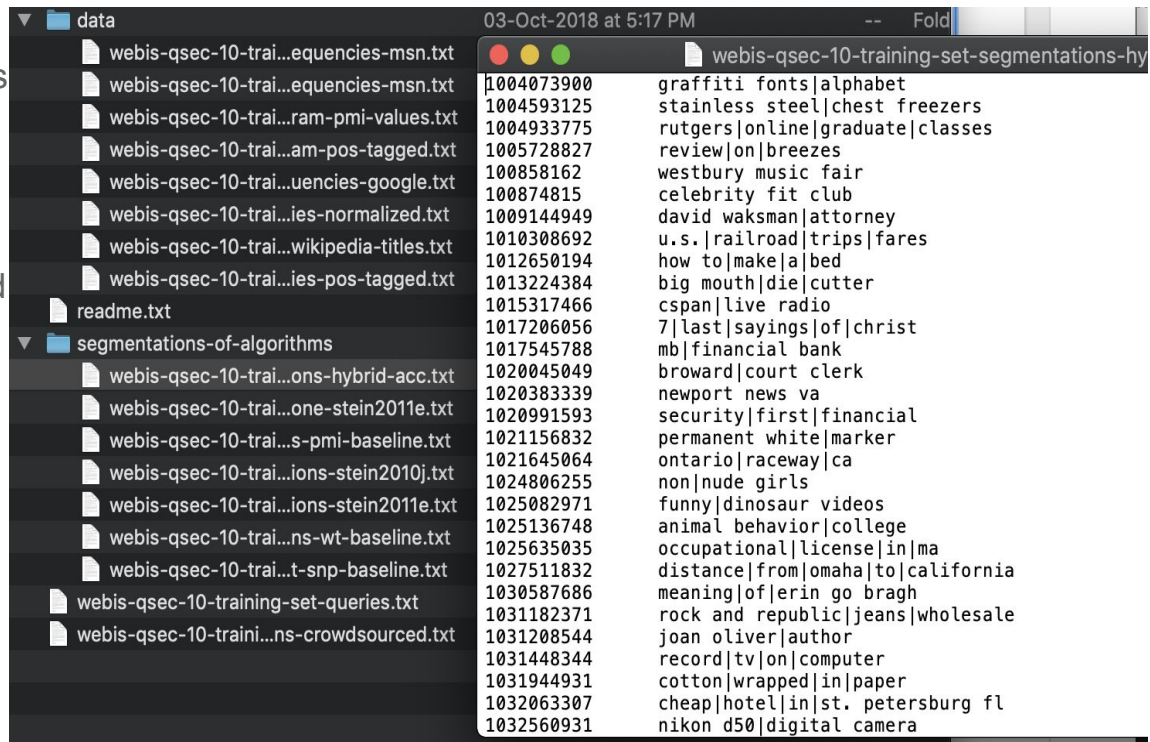
- Labelled Human annotations can lead to very small dataset.
- The IR performance is upper bounded by annotator's skills.
- Lack of intuitive evaluation methods.
- Grammatical structure not as well understood as in sentences in natural language processing.

# Evaluation Framework

- **Need:** Current match based metric(Query Accuracy/ Segment precision) doesn't correlate with IR performance.
- Uses human relevance judgements(RJs) using an IR engine as a black box.
- **Needs the IR engine to support to detect a segment present in the same sequence ordered contiguity match/ unordered sequence match if written in double quotes under linguistic dependence.**
- **Split in brute force set( $2^n - 1$  ways) and find Discounted Cumulative gain for top k positions:  $nDCG@k(q, O_i, R) = r(q, O_{i1}) + \frac{1}{\sum_{j=1}^k (r(q, O_{ij}) / \log_2 j)}$  , Mean average Precision or Mean reciprocal Rank.**
- **Use the  $\max_i(\text{Metric}(q, O_i, R))$  as Oracle score.**
- **QVRS computation:  $QVRS(Q, A, nDCG@k) = 1 - \frac{1}{|Q|} \sum_n (\Omega nDCG@k(q, A))$  - Quoted Version Retrieval Score**

# Dataset

- Three human annotators provided RJs mentioning usefulness of URL given query for all links obtained for all combinations of quoted version of queries.
- The queries were a randomly sampled query log from bing Australia deduplicated and randomized to remove top bias.
- Another dataset webis-qsec provided by amazon mechanical turk.
- The data is in the following forms :  
ID - Query string segmented by | (separator)



The screenshot shows a file explorer window with a sidebar containing a tree view of files. The main pane displays the content of a selected file. The tree view includes a 'data' folder with several text files, a 'segmentations-of-algorithms' folder, and a 'readme.txt' file. The selected file, 'webis-qsec-10-training-set-segmentations-hy', contains a list of IDs and their corresponding segmented query strings.

```
data
├── webis-qsec-10-trai...equencies-msn.txt
├── webis-qsec-10-trai...equencies-msn.txt
├── webis-qsec-10-trai...ram-pmi-values.txt
├── webis-qsec-10-trai...am-pos-tagged.txt
├── webis-qsec-10-trai...uencies-google.txt
├── webis-qsec-10-trai...ies-normalized.txt
├── webis-qsec-10-trai...wikipedia-titles.txt
├── webis-qsec-10-trai...ies-pos-tagged.txt
├── readme.txt
└── segmentations-of-algorithms
    ├── webis-qsec-10-trai...ons-hybrid-acc.txt
    ├── webis-qsec-10-trai...one-stein2011e.txt
    ├── webis-qsec-10-trai...s-pmi-baseline.txt
    ├── webis-qsec-10-trai...ions-stein2010j.txt
    ├── webis-qsec-10-trai...ions-stein2011e.txt
    ├── webis-qsec-10-trai...ns-wt-baseline.txt
    ├── webis-qsec-10-trai...t-snp-baseline.txt
    ├── webis-qsec-10-training-set-queries.txt
    └── webis-qsec-10-traini...ns-crowdsourced.txt
```

03-Oct-2018 at 5:17 PM -- Fold

```
webis-qsec-10-training-set-segmentations-hy
1004073900 graffiti fonts|alphabet
1004593125 stainless steel|chest freezers
1004933775 rutgers|online|graduate|classes
1005728827 review|on|breezes
100858162 westbury music fair
100874815 celebrity fit club
1009144949 david waksman|attorney
1010308692 u.s.|railroad|trips|fares
1012650194 how to|make|a|bed
1013224384 big mouth|die|cutter
1015317466 cspan|live radio
1017206056 7|last|sayings|of|christ
1017545788 mb|financial bank
1020045049 broward|court clerk
1020383339 newport news va
1020991593 security|first|financial
1021156832 permanent white|marker
1021645064 ontario|raceway|ca
1024806255 non|nude girls
1025082971 funny|dinosaur videos
1025136748 animal behavior|college
1025635035 occupational|license|in|ma
1027511832 distance|from|omaha|to|california
1030587686 meaning|of|erin go bragh
1031182371 rock and republic|jeans|wholesale
1031208544 joan oliver|author
1031448344 record|tv|on|computer
1031944931 cotton|wrapped|in|paper
1032063307 cheap|hotel|in|st. petersburg fl
1032560931 nikon d50|digital camera
```

# Work Done so far

We implemented the following well known algorithms on that data.

- **Baseline Methods**

- Wikipedia titles + strict noun phrases.
- Conditional random fields (models  $P(\text{labels} \mid \text{query})$ ).
- Web n-gram probabilities(PMI-W)
- Query logs(PMI-Q)

- **Experiments**

- Unsupervised query segmentation using clickthrough for information retrieval using expectation maximization (Li et al)
- Unsupervised query segmentation using only query logs (Mishra et al).
- RNN - Autoencoder decoder on Webis-qsec data.

# Results–Bing+HA

Table 3: Results of IR-based evaluation of segmentation algorithms using Lucene (mean oracle scores).

Metric	Unseg. query	[9]	[7]	[11]	[11] + Wiki	PMI-W	PMI-Q	$H_A$	$H_B$	$H_C$	$BQV_{BF}$
nDCG@5	0.688	0.752*	<b>0.763*</b>	0.745	<b>0.767*</b>	0.691	<b>0.766*</b>	<b>0.770</b>	0.768	0.759	0.825
nDCG@10	0.701	0.756*	<b>0.767*</b>	0.751	<b>0.768*</b>	0.704	<b>0.767*</b>	<b>0.770</b>	<b>0.768</b>	<b>0.763</b>	0.832
MAP@5	0.882	0.930*	<b>0.942*</b>	0.930*	<b>0.945*</b>	0.884	0.932*	<b>0.944</b>	<b>0.942</b>	0.936	0.958
MAP@10	0.865	0.910*	<b>0.921*</b>	0.910*	<b>0.923*</b>	0.867	0.912*	<b>0.923</b>	<b>0.921</b>	<b>0.916</b>	0.944
MRR@5	0.538	0.632*	<b>0.649*</b>	0.609	<b>0.650*</b>	0.543	<b>0.648*</b>	<b>0.656</b>	<b>0.648</b>	0.632	0.711
MRR@10	0.549	0.640*	<b>0.658*</b>	0.619	<b>0.658*</b>	0.555	<b>0.656*</b>	<b>0.665</b>	<b>0.656</b>	0.640	0.717

The highest value in a row (excluding the  $BQV_{BF}$  column) and those with no statistically significant difference with the highest value are marked in **boldface**. The values for algorithms that perform better than or have no statistically significant difference with the *minimum* of the human segmentations are marked with \*. The paired *t*-test was performed and the null hypothesis was rejected if the *p*-value was less than 0.05.

Table 4: Matching metrics for different segmentation algorithms and human annotations *with  $BQV_{BF}$  as reference*.

Metric	Unseg. query	[9]	[7]	[11]	[11] + Wiki	PMI-W	PMI-Q	$H_A$	$H_B$	$H_C$	$BQV_{BF}$
Qry-Acc	0.044	0.056	0.082*	0.058	0.094*	0.046	<b>0.104*</b>	0.086	0.074	0.064	1.000
Seg-Prec	<b>0.226*</b>	0.176*	0.189*	0.206*	0.203*	<b>0.229*</b>	0.218*	0.176	0.166	0.178	1.000
Seg-Rec	<b>0.325*</b>	0.166*	0.162*	0.210*	0.174*	<b>0.323*</b>	0.196*	0.144	0.133	0.154	1.000
Seg-F	<b>0.267*</b>	0.171*	0.174*	0.208*	0.187*	<b>0.268*</b>	0.206*	0.158	0.148	0.165	1.000
Seg-Acc	0.470	0.624	0.661*	0.601	0.667*	0.474	0.660*	<b>0.675</b>	<b>0.675</b>	0.663	1.000

The highest value in a row (excluding the  $BQV_{BF}$  column) and those with no statistically significant difference with the highest value are marked in **boldface**. The values for algorithms that perform better than or have no statistically significant difference with the *minimum* of the human segmentations are marked with \*. The paired *t*-test was performed and the null hypothesis was rejected if the *p*-value was less than 0.05.

Table 6: Kendall-Tau coefficients between IR :  
matching metrics *with  $BQV_{BF}$  as reference for latter*.

Metric	Qry-Acc	Seg-Prec	Seg-Rec	Seg-F	Seg-Acc
nDCG@10	0.432	-0.854	-0.886	-0.854	<b>0.674</b>
MAP@10	0.322	-0.887	-0.920	-0.887	<b>0.750</b>
MRR@10	0.395	-0.782	-0.814	-0.782	<b>0.598</b>

The highest value in a row is marked in **boldface**.

# Results– Webis–qsec

	query accuracy	break accuracy
WT [7]	0.431	0.769
WT+SNP [7]	0.585	0.837
CRF	0.465	0.814
RNN Encoder-Decoder	0.421	0.664

**Table 2: Query level and break level accuracy on Webis-QSeC-10 test set.**



# Inference & Our Approach

## Inference

- Existing methods(except RNNs) are upper bounded by underlying IR engine's ranking and model doesn't remember the Importance of which sequence of words together leads to better results to utilize in other queries.
- The metric to obtain oracle scores don't penalize for splitting more or calculate any information loss while doing so.
- Adding separator to Vocabulary in RNNs is misleading since It tries to memorize which sequence of words generally occur before or after a separator.

## Approach - Hybrid Model design

- Instead of Adding separator to Vocabulary in RNNs , we can use RNNs to learn which sequence of words are good enough to form a segment by giving a one hot vector sequence of words and treating the input as vector of words for train and skip until next word if a separator occurs.
- Adding discounted cumulative gain for that segment to loss function before applying Adam optimizer on it makes results more IR performance relevant.
- GRU/LSTM unit with appropriate parameter set can be even more effective if the queries are large.

# Future Work left to do

- Implement hybrid Model and compare both matching score/IR based evaluation framework scores
- Design unsupervised ML based method for good generalization.
- Design better evaluation criteria/models that capture the essence of word sequence-relevance relationship given query to avoid overfitting to the annotated segment.

# References

- An IR-based Evaluation Framework for Web Search Query Segmentation by Rishiraj Saha Roy and Niloy Ganguly
- Query Segmentation via RNNs Encoder-Decoder Framework