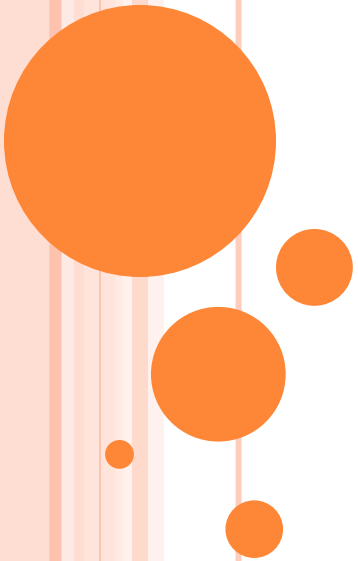
DivText

TEXT RECOMMENDATION SYSTEM BASED ON RESULT DIVERSIFICATION



Bahaeddin ERAVCI
Emre YILMAZ
Izzeddin GUR
Mehmet GUVERCIN

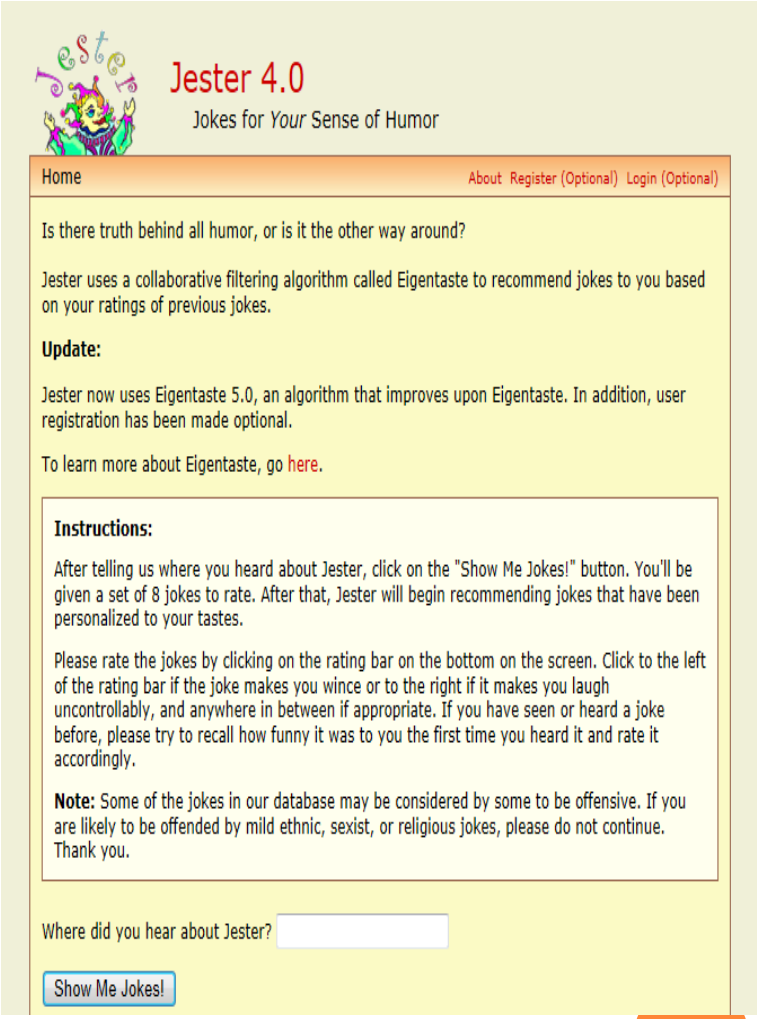
Agenda

- Problem: diverse text recommendation
- Motivation
- Applied methodology
- Experiments on Jester Joke dataset
- Experimental results

Problem Description

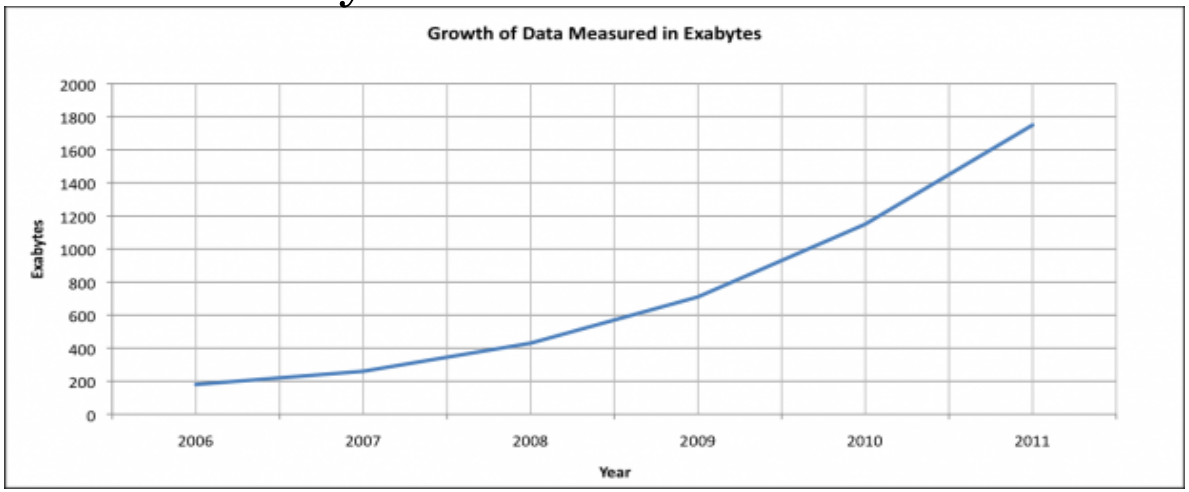
- We have user queries, and we want to find similar but diverse documents to offer

- There is a tradeoff between similar and diverse documents



Motivation

- Big data
  - The fastest increasing quantity on this planet is the amount of information we are generating
- Large availability
- Partial knowledge about data
  - if you don't know the alternatives you don't know to search for exactly



Motivation

- So, we need results that are both *similar* to the query yet *different* from each other i.e. **diversity**

- This gives a chance
  - to do exploratory search
  - see different perspectives of the query
  - for a better user satisfaction

Methodology

- Given a user profile
- Goal: find similar but diverse k documents for each user

Steps of methodology:

1) Extract document-term matrix
2) Generate user profiles
3) Get top k documents from diverse system
4) Evaluate results based on the user satisfaction

Methodology: Step 1

- Clearing punctuation marks
- Removing stop words
- Stemming
  - Tartarus Snowball
- Removing unnecessary words
- Constructing document-term frequency matrix
- 1115 terms – 100 documents

Methodology: Step 2

- To extract queries for users
  - Get vector of top m documents that user most likes
  - Take the union of document vectors

$$q = \bigcup_{i=1}^{m} d_i$$

Methodology: Step 3

- Calculate $s(q, d_i)$ for all i
  - $s(q, d_i) \stackrel{\text{def}}{=} \frac{2 * q \cap d_i}{|q| + |d_i|}$
- Put most similar one into RelDocs
- Calculate how much the document d_i is diverse
  - $div(di, RelDocs) \stackrel{\text{def}}{=} \min\left[1 - s(d_i, d_j)\right] \quad \forall\, dj \in RelDocs$
- Calculate the relevance of the document d_i
  - $\text{Rel}(d_i, \text{q}, \text{RelDocs}) = \alpha * \text{s}(\text{q}, d_i) + (1 - \alpha) * \text{div}(d_i, \text{RelDocs})$

Methodology: Step 3

- α is the parameter to adjust similarity and diversity
- If α = 1, then only similar documents will be recommended

- Put document with maximum Rel into RelDocs
- Stop after k iterations
- Report RelDocs

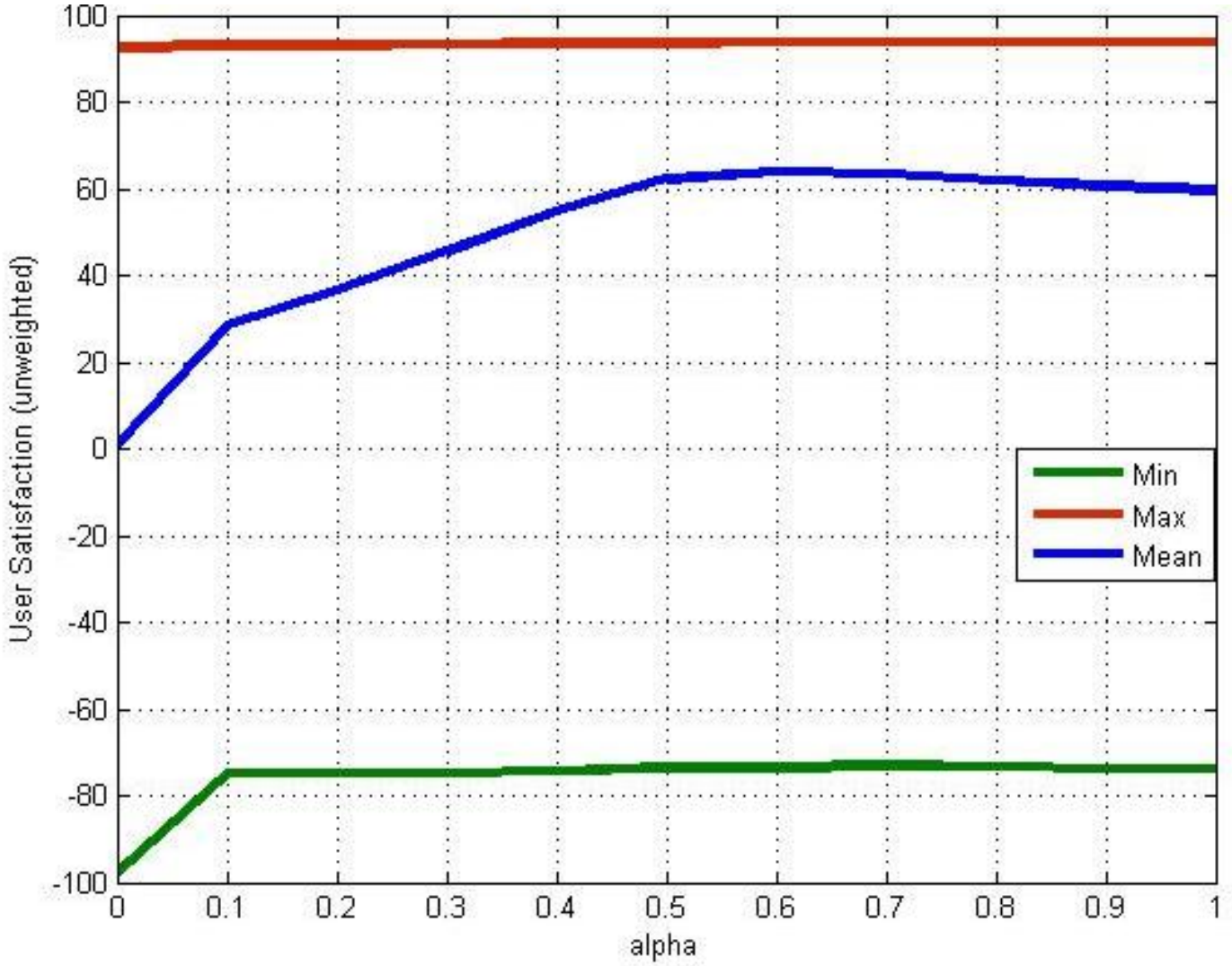
Methodology: Step 4

- At the end of the step 3 we have k recommended documents
- Evaluate user satisfaction(unweighted)
  - Sum user ratings for k documents
  - Without ranking
- Evaluate user satisfaction(weighted)
  - ∑(user rating) *(reverse rank)
  - Reverse rank = k – rank + 1

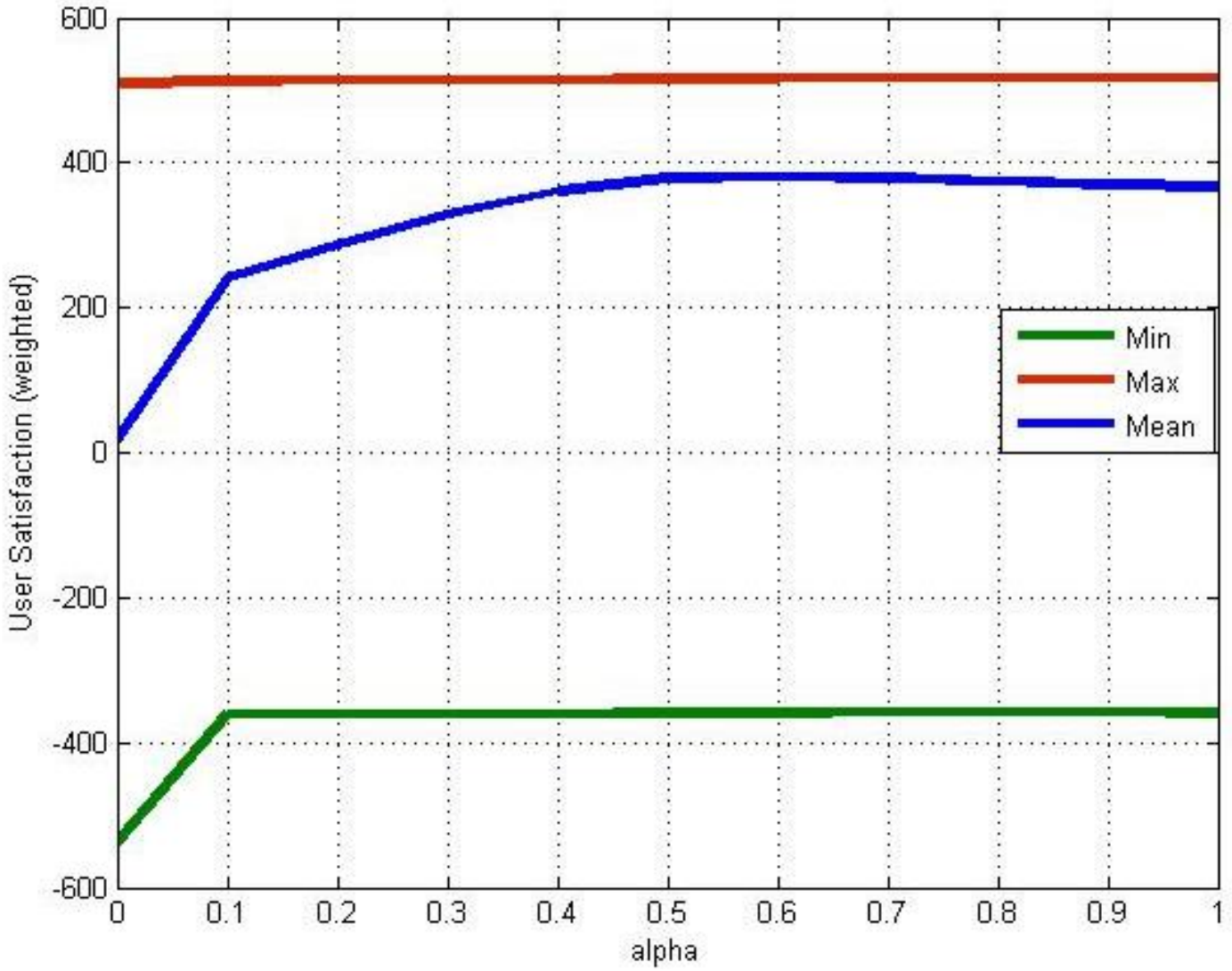
Experimental Setup

- Dataset:
  - 100 jokes are rated by 7200 users with scale [-10,10]
- α is incremented from 0.0 to 1.0 by 0.1
- Min, mean, max satisfaction is calculated for each α
- k = 10
- m = 10

Experimental Results



Experimental Results



Conclusion

- Diverse results give better user satisfaction with respect to similarity search
  - User satisfaction increases 20x
- User satisfaction increases until an optimum alpha value
  - Optimum alpha value is found 0.6 for this dataset
- Diversity even works for the most unsatisfied users

Future work

- Recommendation system based on collaborative filtering
- Learn α for each user
- Using clustering analysis for diversity

Q & A