Document Classification

based on

LDA

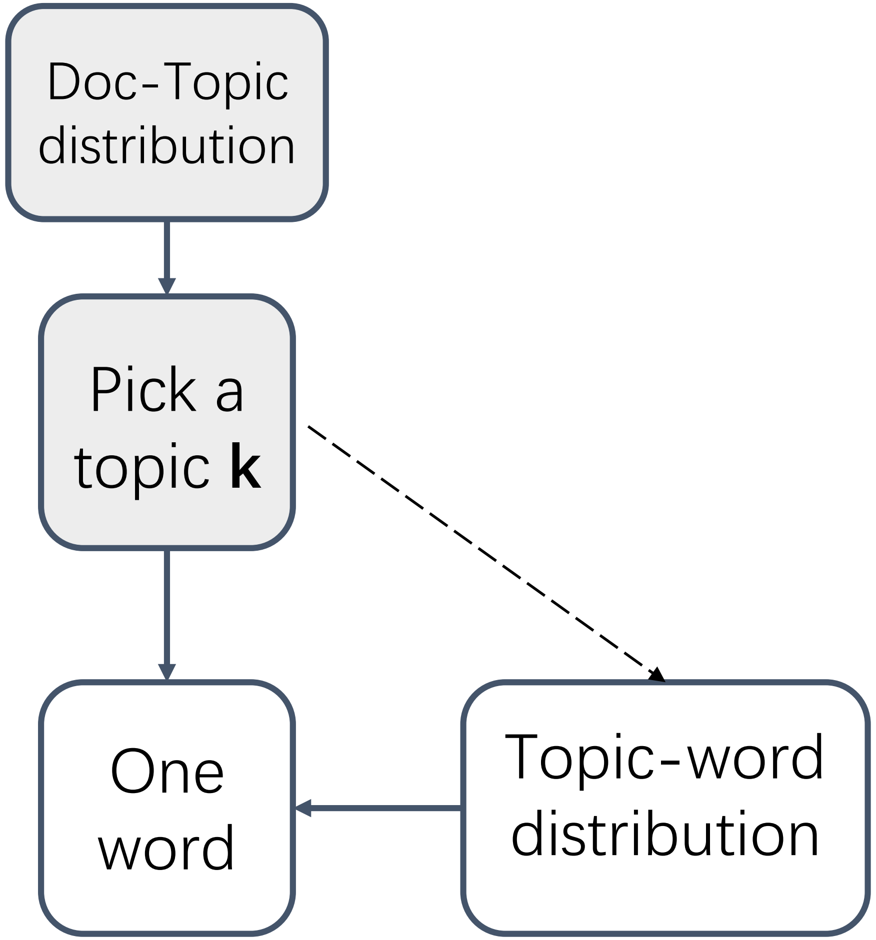
Latent Dirichlet Allocation

Chunyuan Li
Jiarui Xing

Contents

- Introduction to LDA
- Experiment Design
- Experiment Result

LDA Document Generation Model (Simplified)



First Step:

choose a topic from document-topic distribution

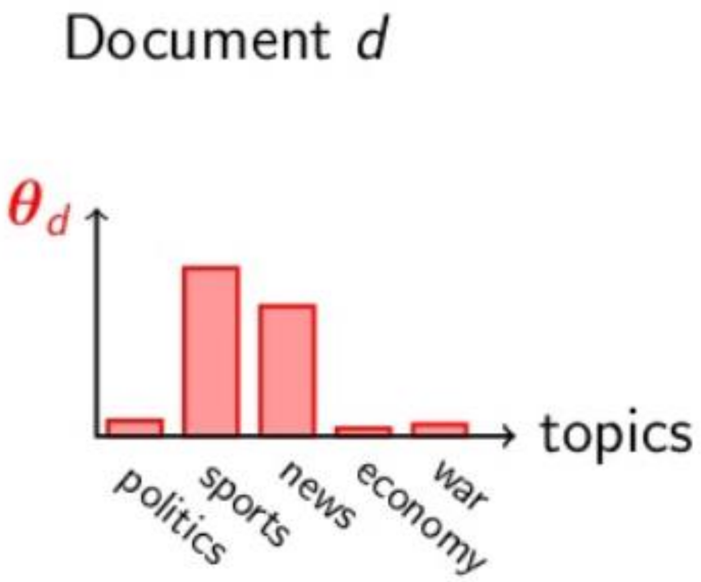
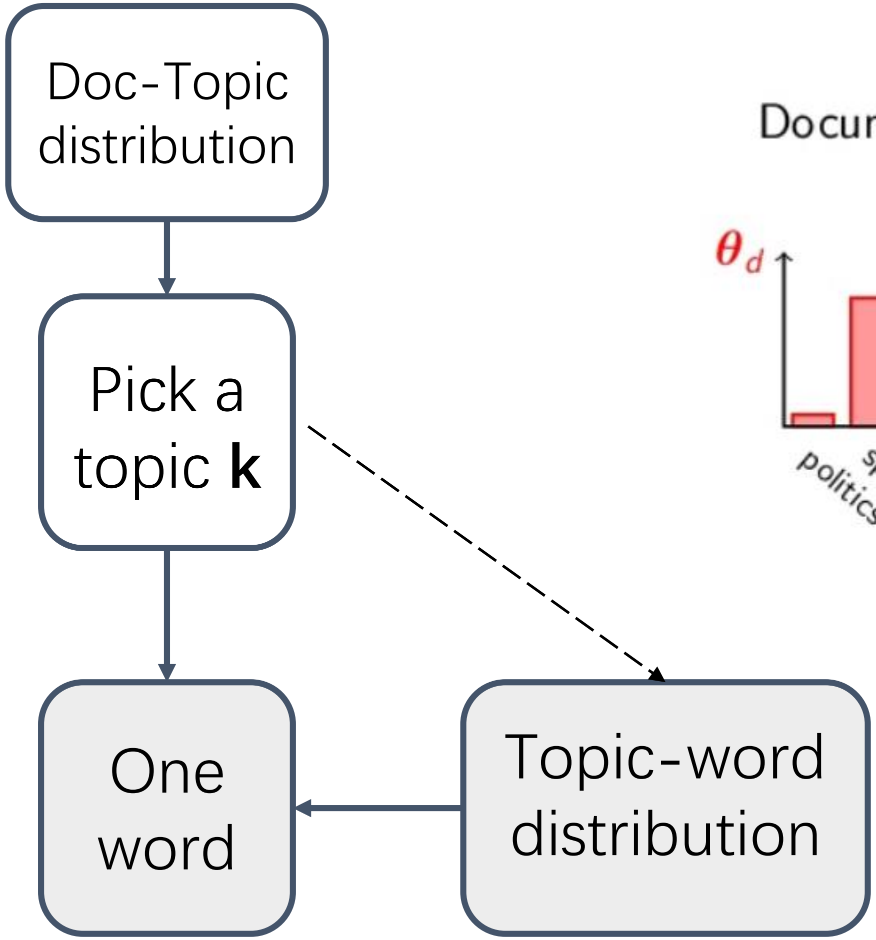
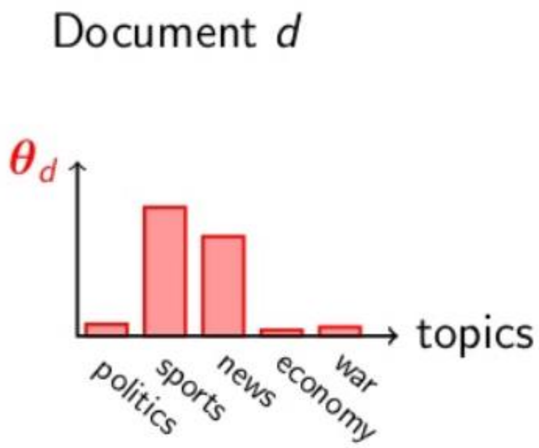
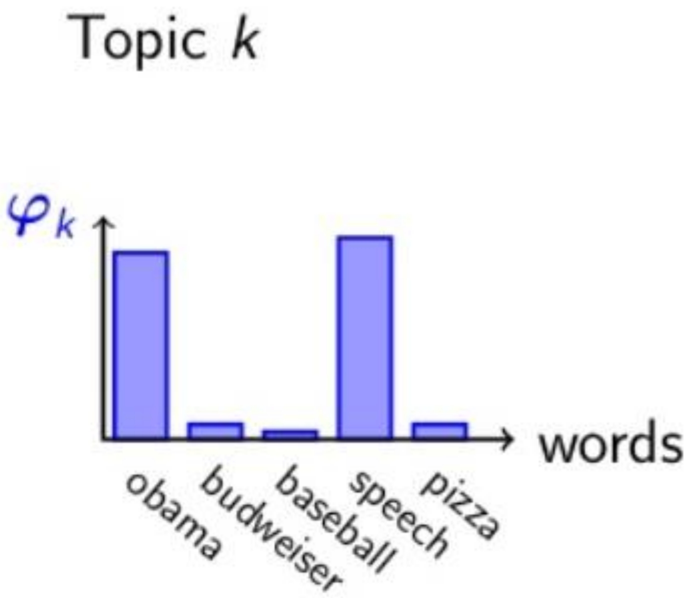


Figure: datasciencecentral.com/profiles/blogs/a-tale-about-lda2vec-when-lda-meets-word2vec

LDA Document Generation Model (Simplified)



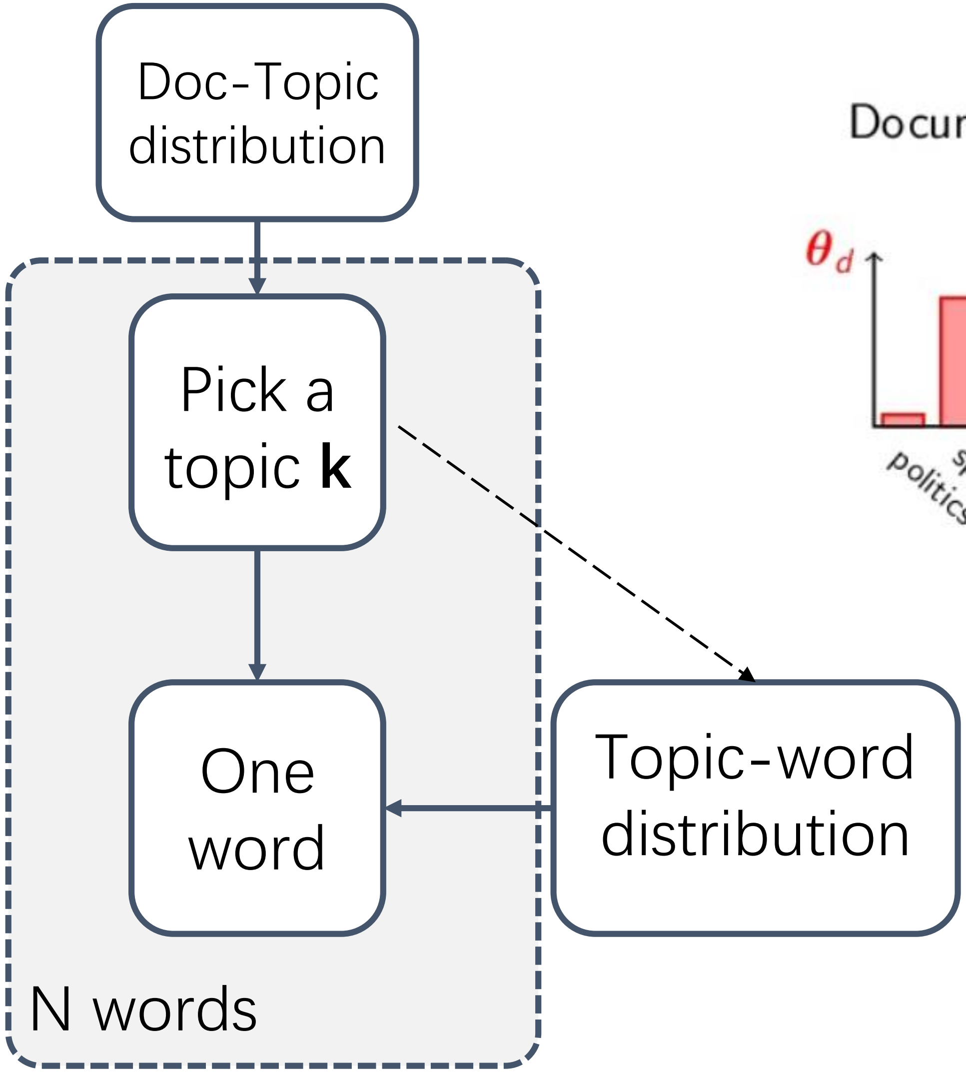


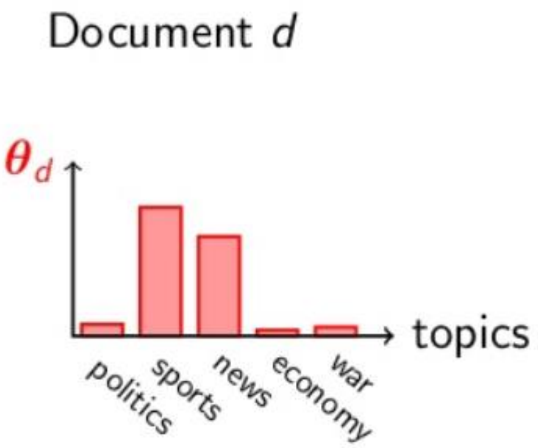


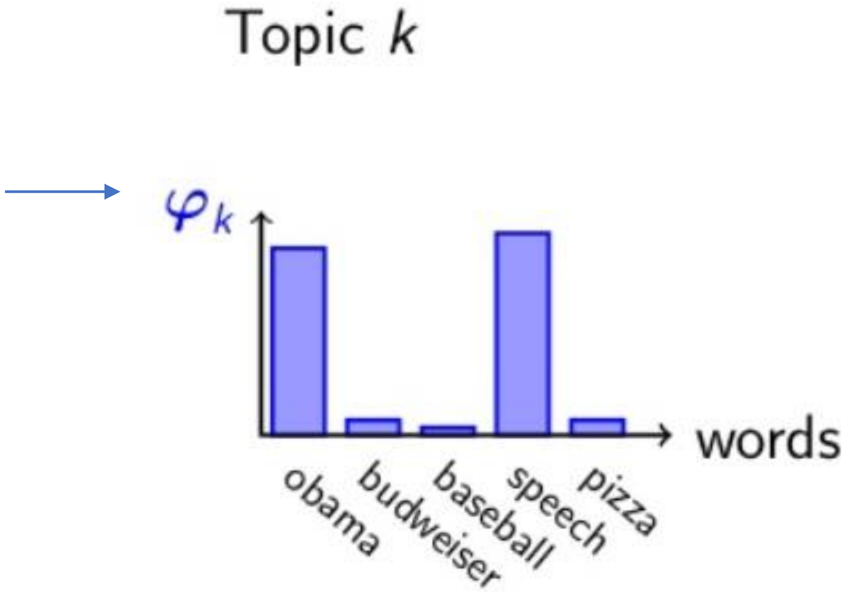
Second Step:

choose a word from topic-word distribution

LDA Document Generation Model (Simplified)

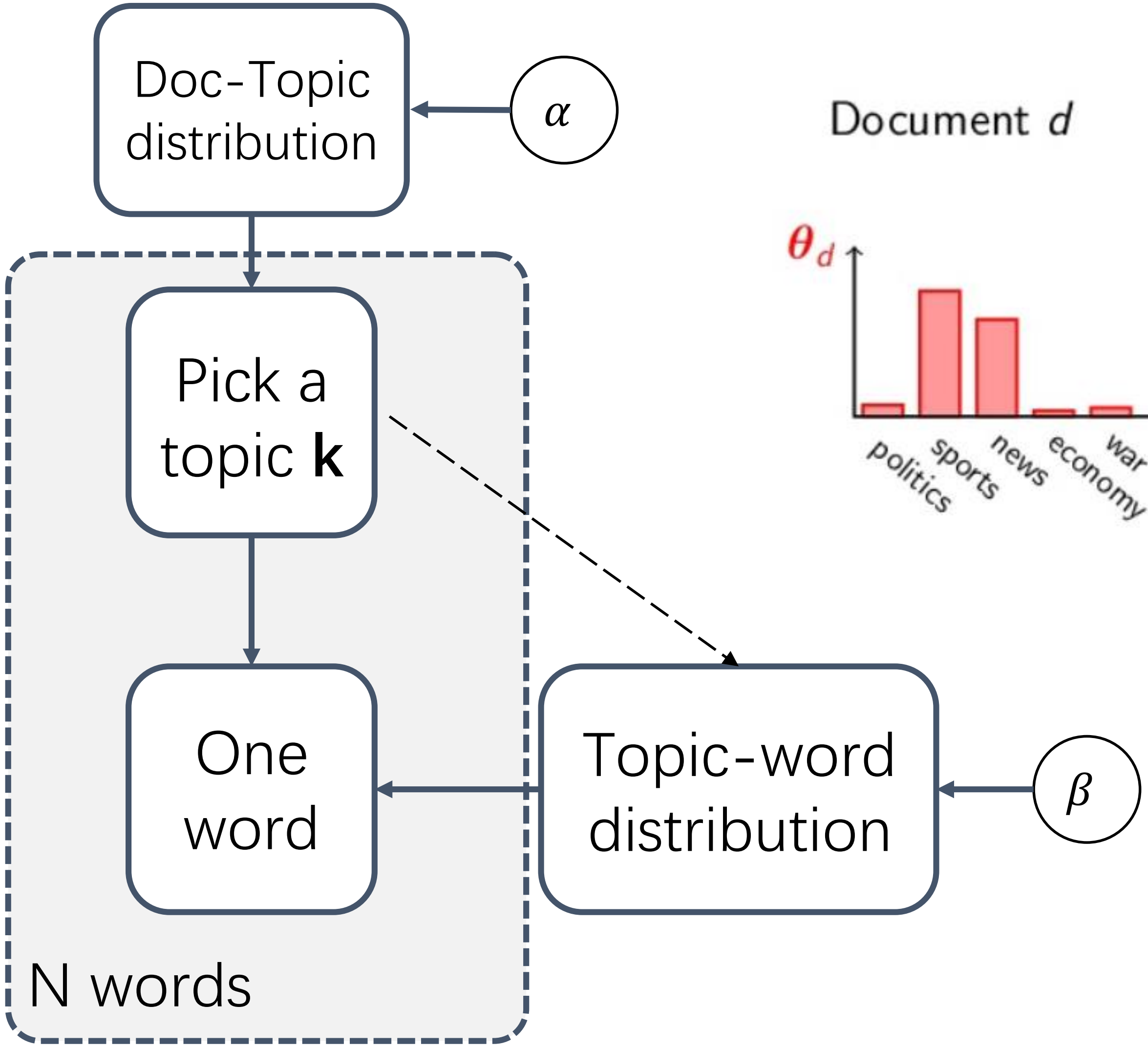


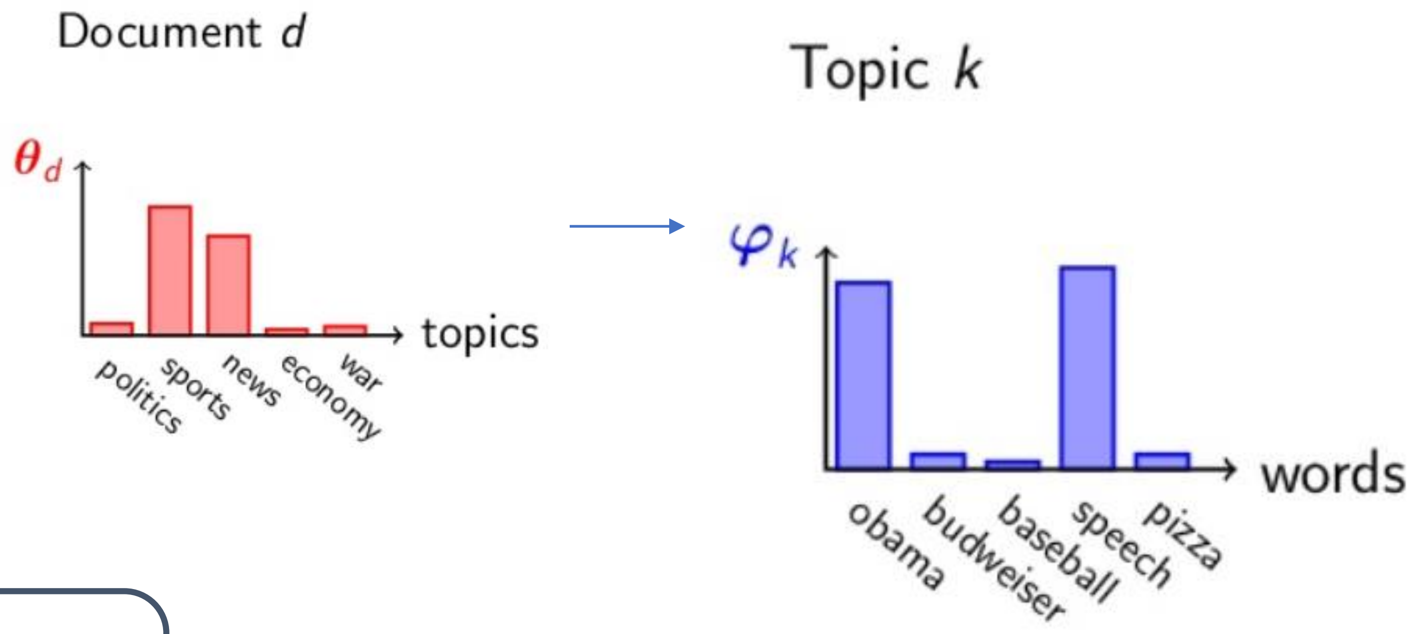




Do it N times to generate a document with N words

LDA Document Generation Model (Simplified)

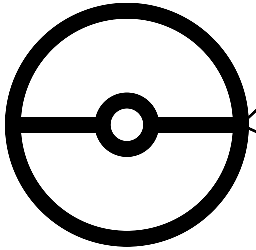




Document Classification Based on LDA

- **Basic Idea**

As different documents have difference in the usage of words (e.g. term frequency/tf-idf), they should also **differ in topics**!



Wikipedia: Pokémon

words → game, video, Nintendo, Japan, ⋯

topics → {Nintendo, game, video,⋯} {media, franchise, company,⋯}

Wikipedia: Bayesian Inference

words → statistical, math, inference, ⋯

topics → {Bayesian, statistical, inference} {prior, posterior, use,⋯}

Document Classification Based on LDA

- **Basic Idea**

  As different documents have difference in the usage of words (e.g. term frequency or tf-idf), they should also differ in topics!

- **Implementation: LDA as Vectorization!**

  Transform a document into **distribution of topics**

Document

↓

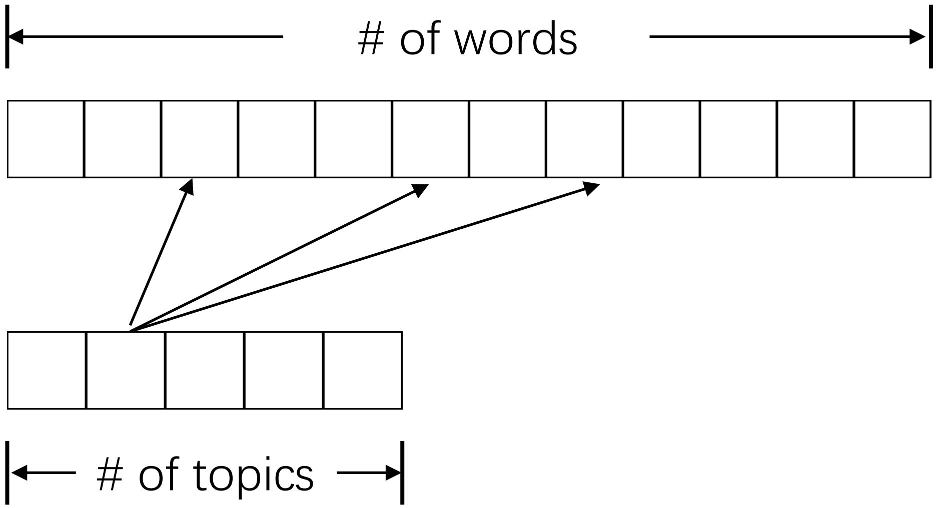
0.12	0.08	0.01	0.23	···	0.01	0.01	0.02	0.23	0.13

Topic Distribution

Experiment Design

- **When should LDA work better (than other document vectorization methods)?**
  - With lower vector dimension?

    Each topic referring to several words, which may represent a document in lower dimensions (than tf and tf-idf).



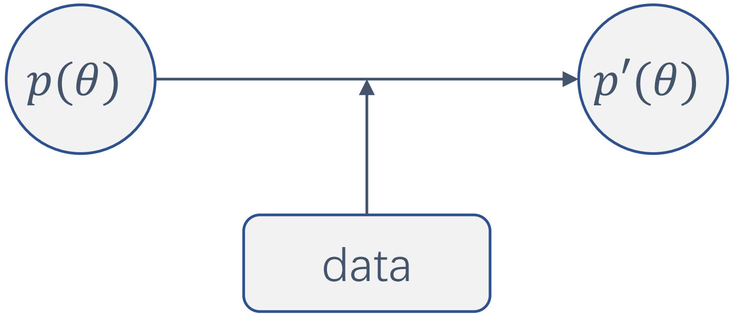
Experiment Design

- **When should LDA work better (than other document vectorization methods)?**
  - With lower vector dimension?

    Each topic referring to several words, which may represent a document in lower dimensions (than tf and tf-idf).

  - With small dataset?

    Prior helps us avoid overfitting (think of pseudo-count in coin flipping case)



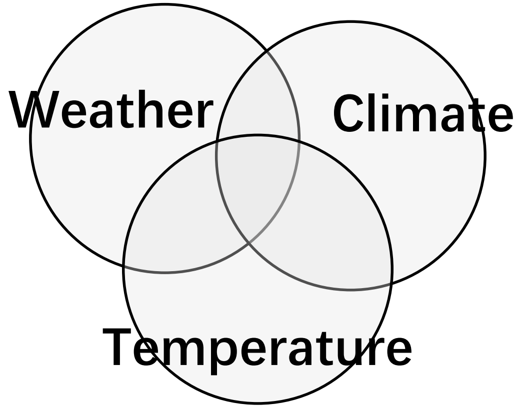
Experiment Design

- **When should LDA work better (than other document vectorization methods)?**
  - With lower vector dimension?

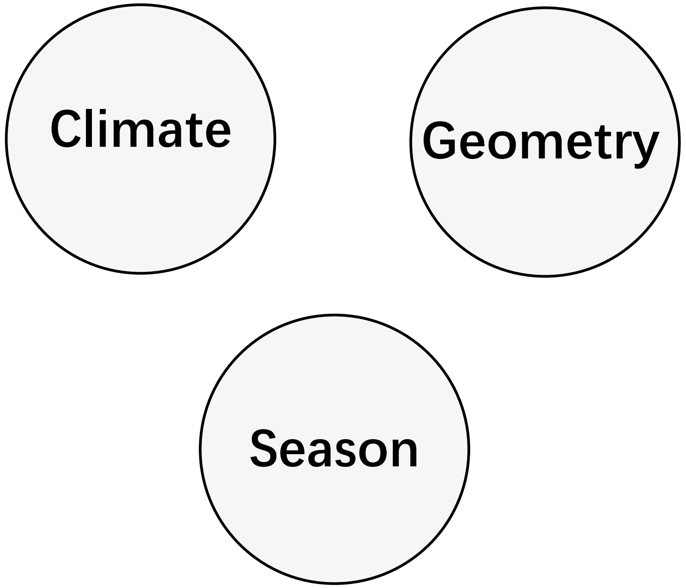
    Each topic referring to several words, which may represent a document in lower dimensions (than tf and tf-idf).
  - With small dataset?

    Prior helps us avoid overfitting(think of pseudo-count in coin flipping case)
  - With smaller document-topic-prior (α)?

    "Sparse" topics work better



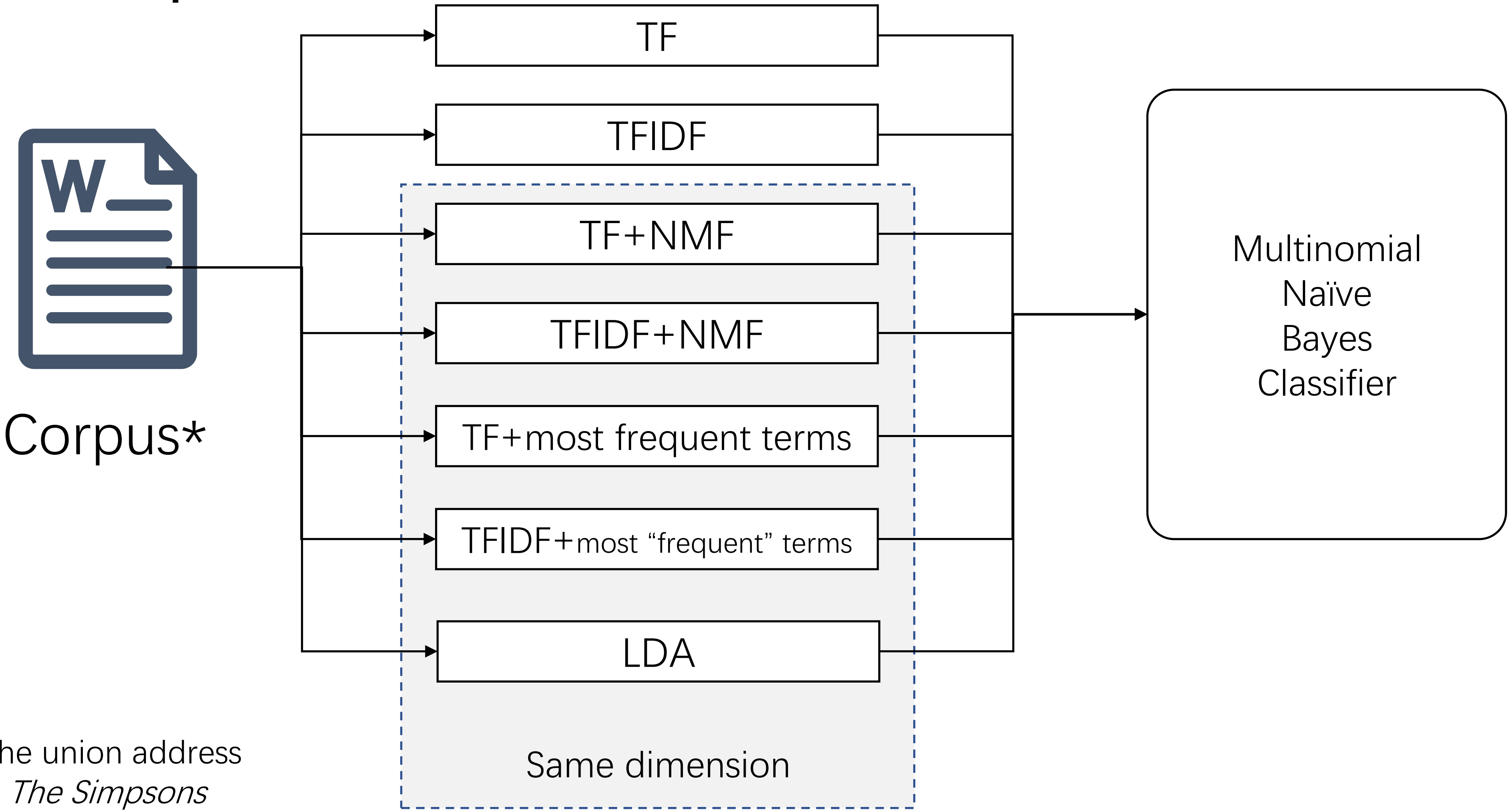
Large α



Small α

Experiment Design

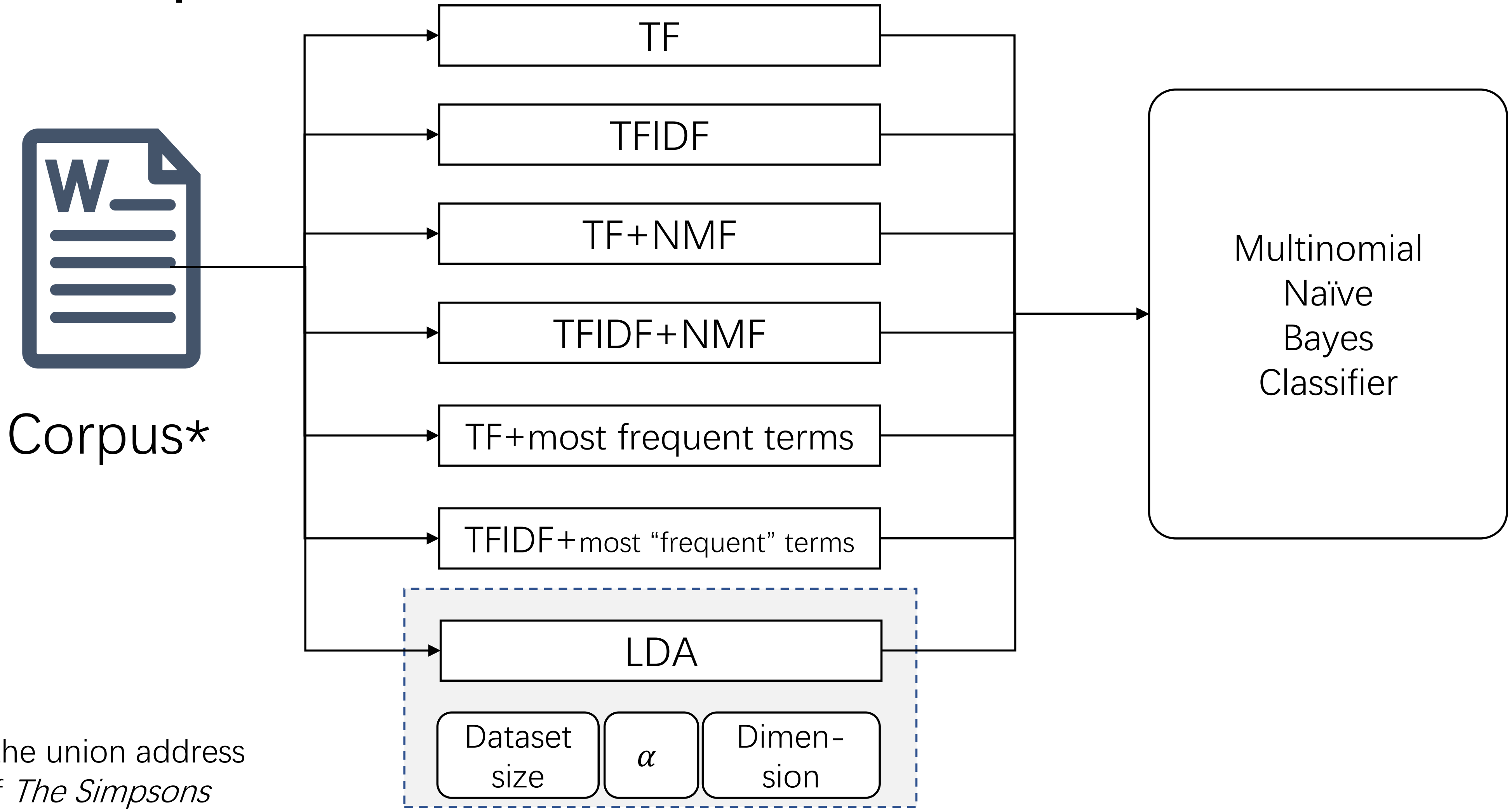
- **Experiment Pipeline**



* 1. State of the union address
2. Scripts of *The Simpsons*

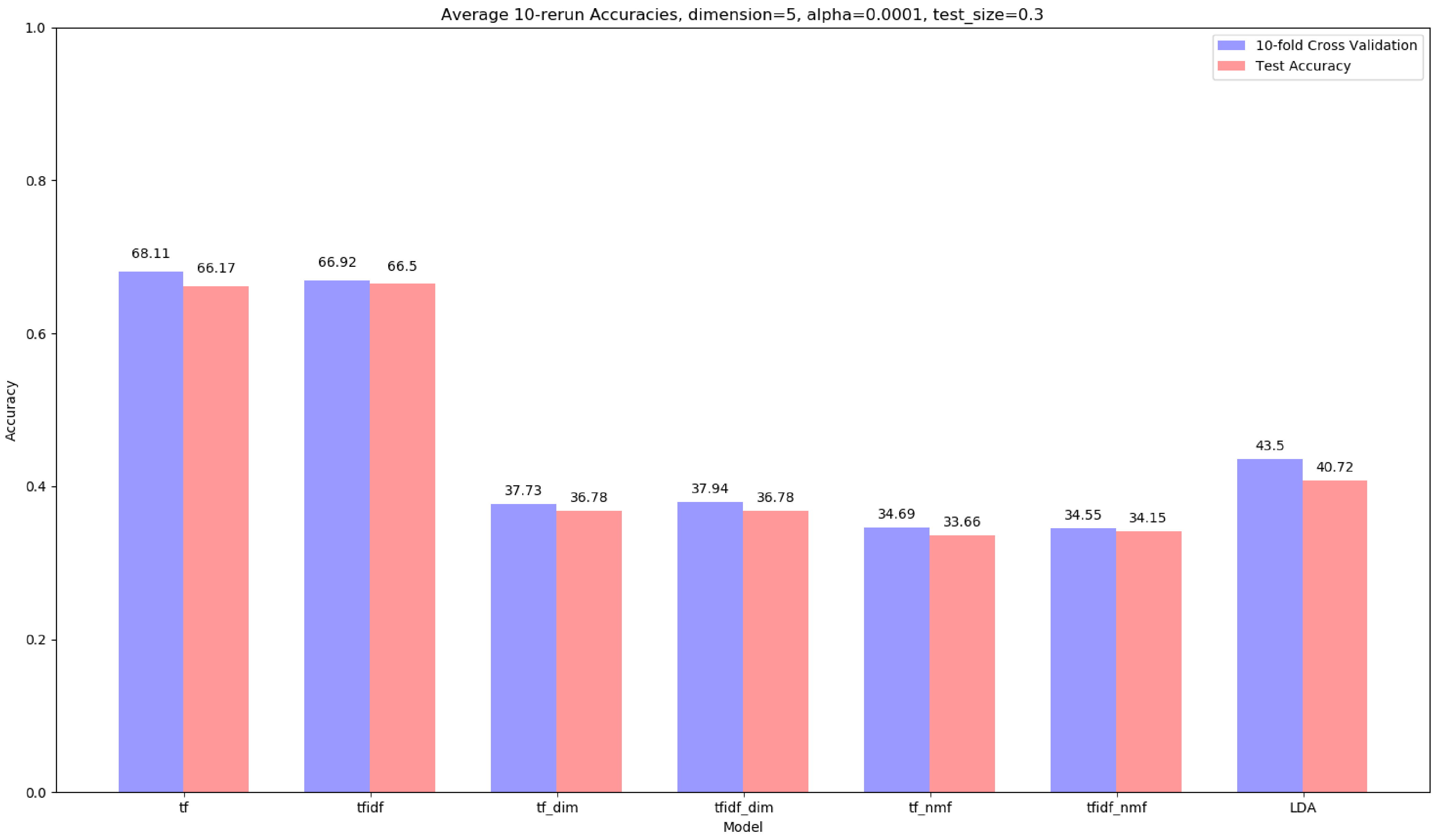
Experiment Design

- **Experiment Pipeline**

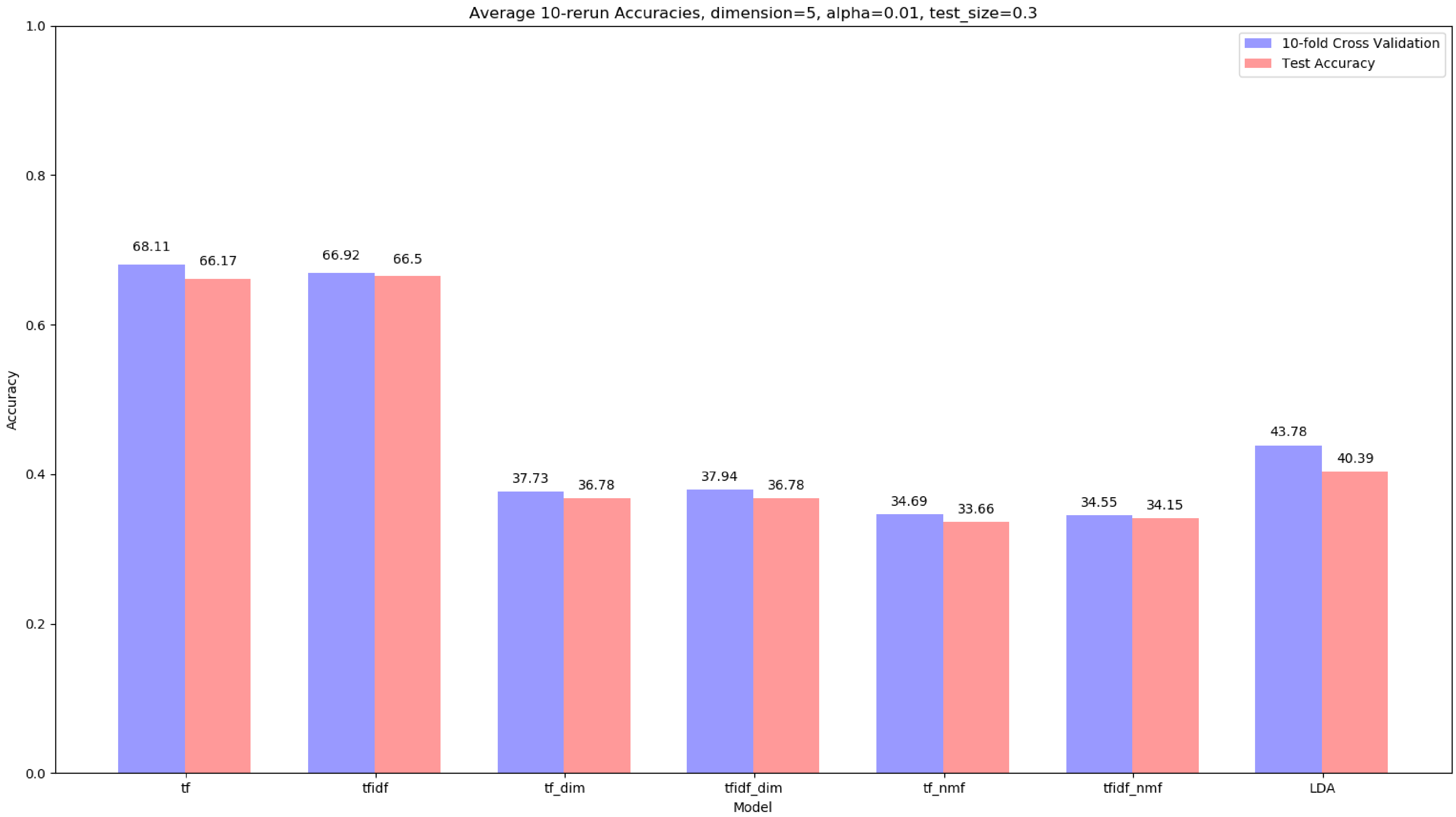


* 1. State of the union address
2. Scripts of *The Simpsons*

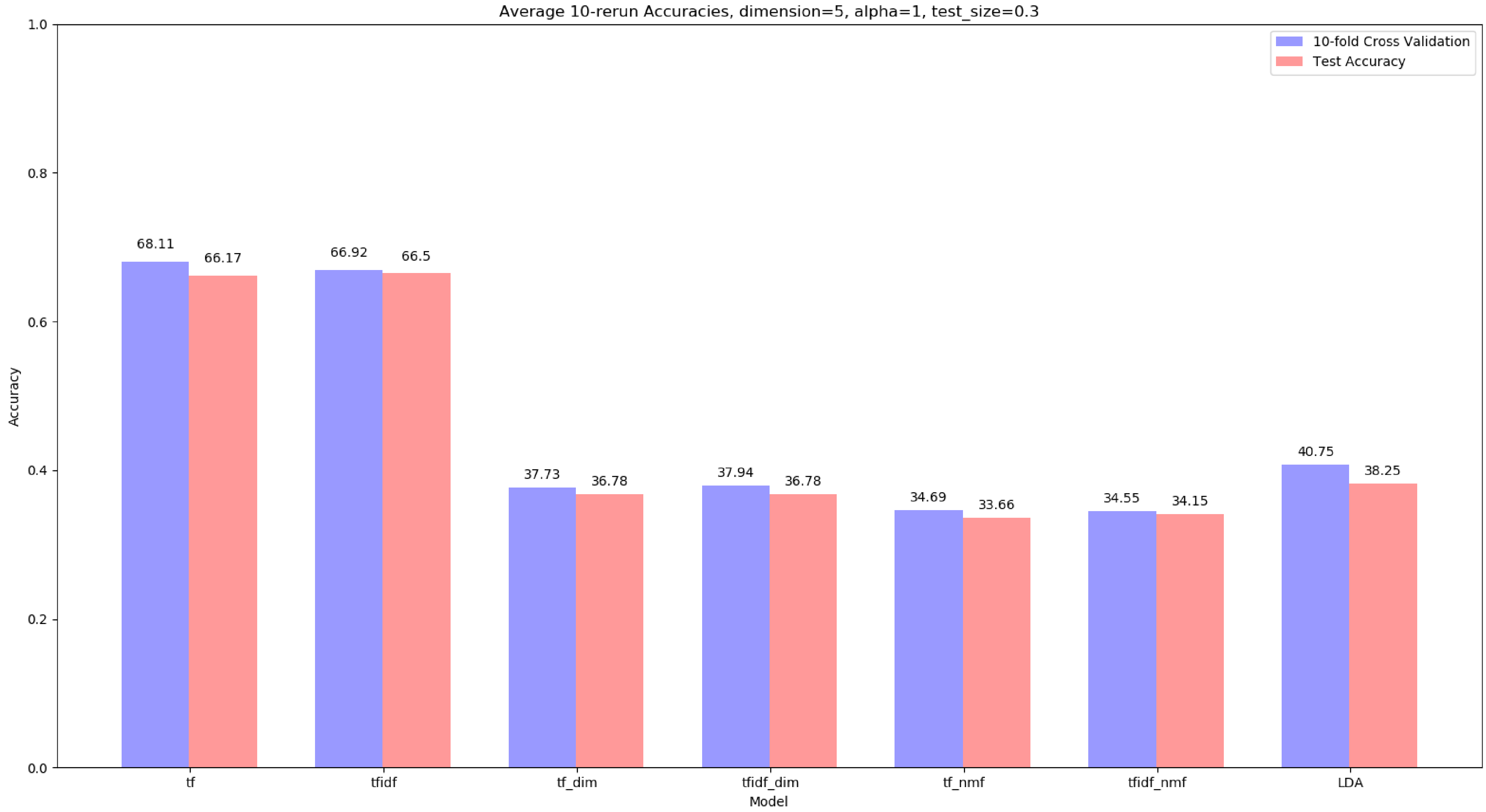
Experiment Result Accuracies with $\alpha = \mathbf{0.0001}, 0.01, 1, 10$



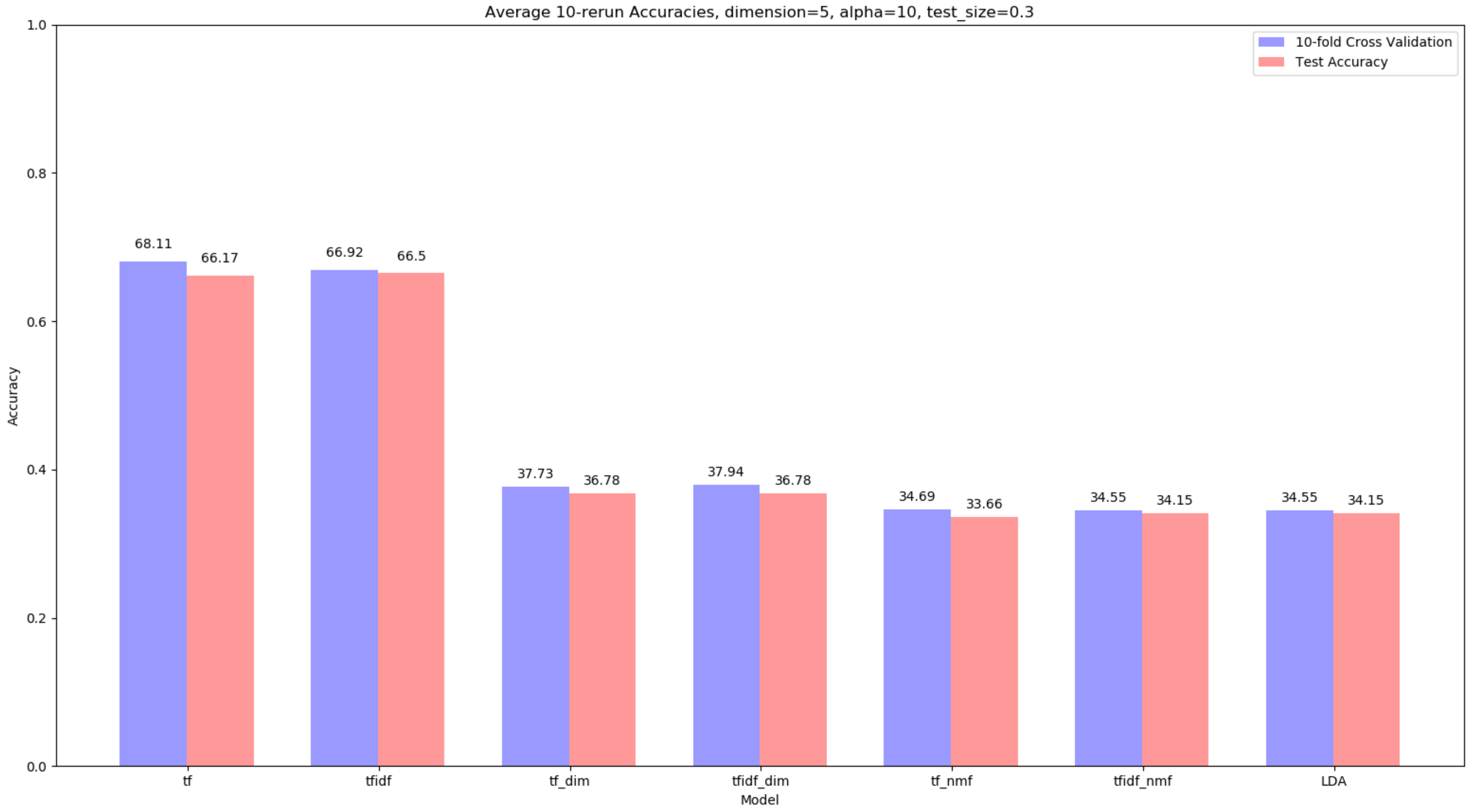
Experiment Result Accuracies with $\alpha = 0.0001, \mathbf{0.01}, 1, 10$



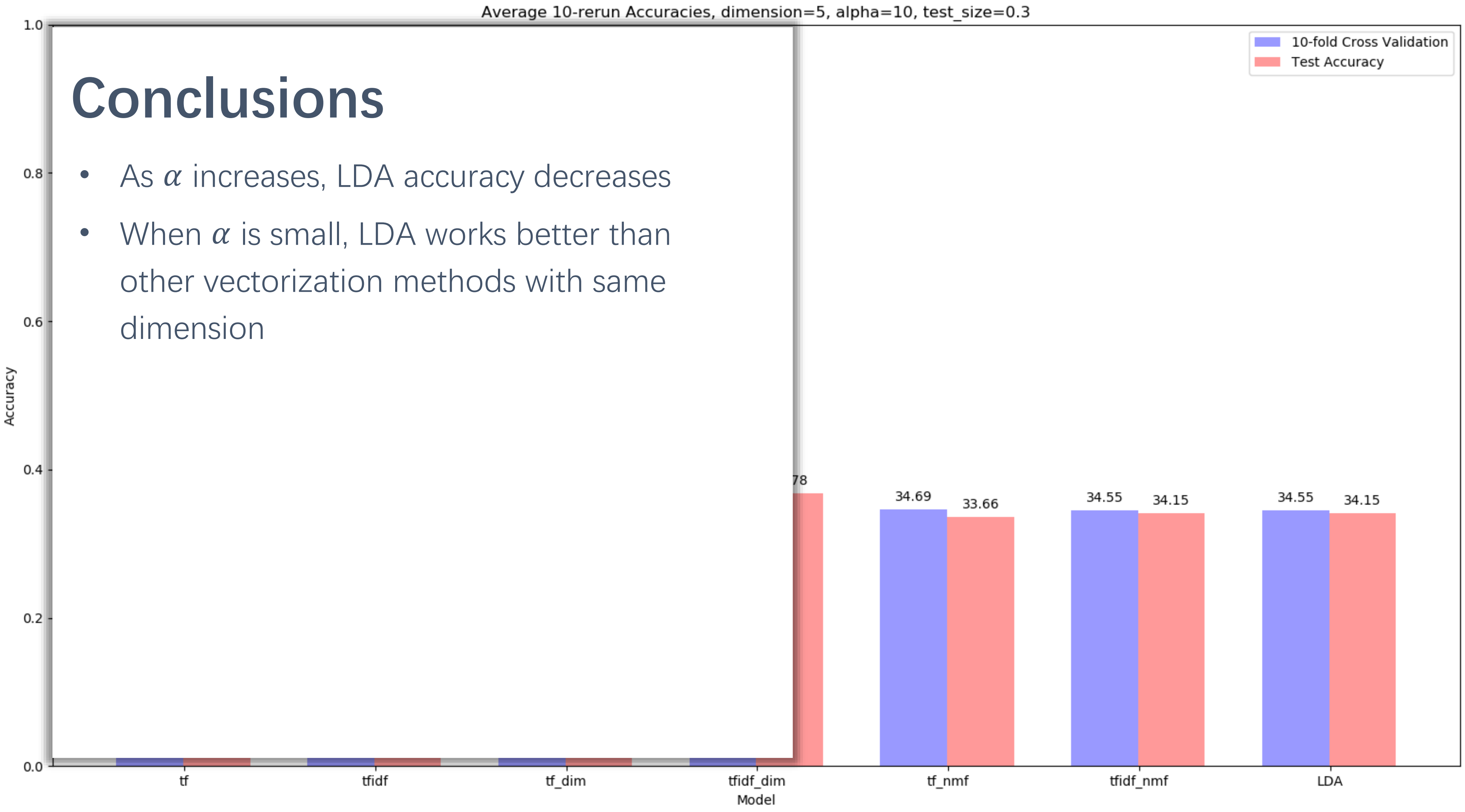
Experiment Result Accuracies with $\alpha = 0.0001, 0.01, \mathbf{1}, 10$



Experiment Result Accuracies with $\alpha = 0.0001, 0.01, 1, \mathbf{10}$



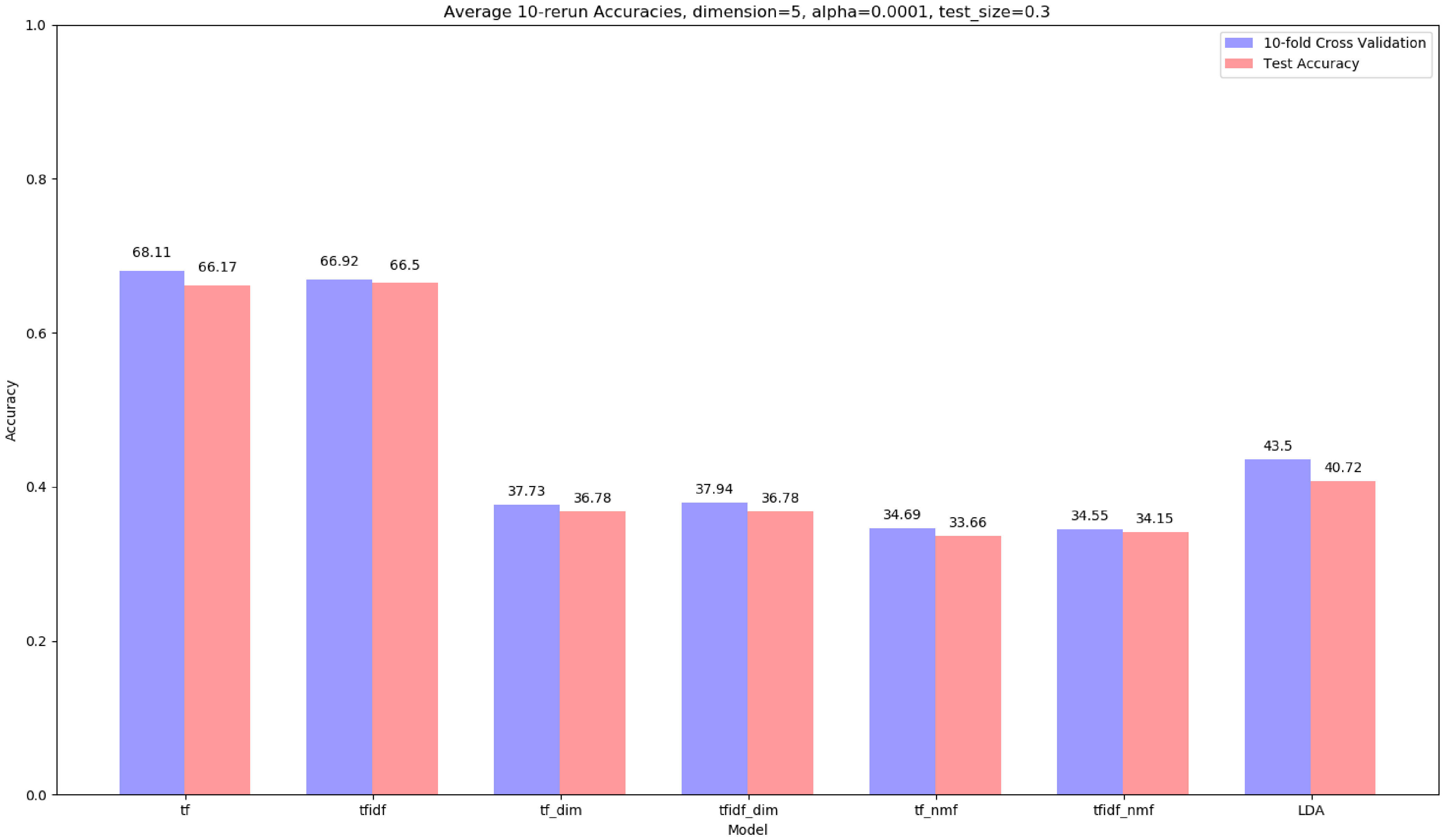
Experiment Result Accuracies with $\alpha = 0.0001, 0.01, 1, \mathbf{10}$



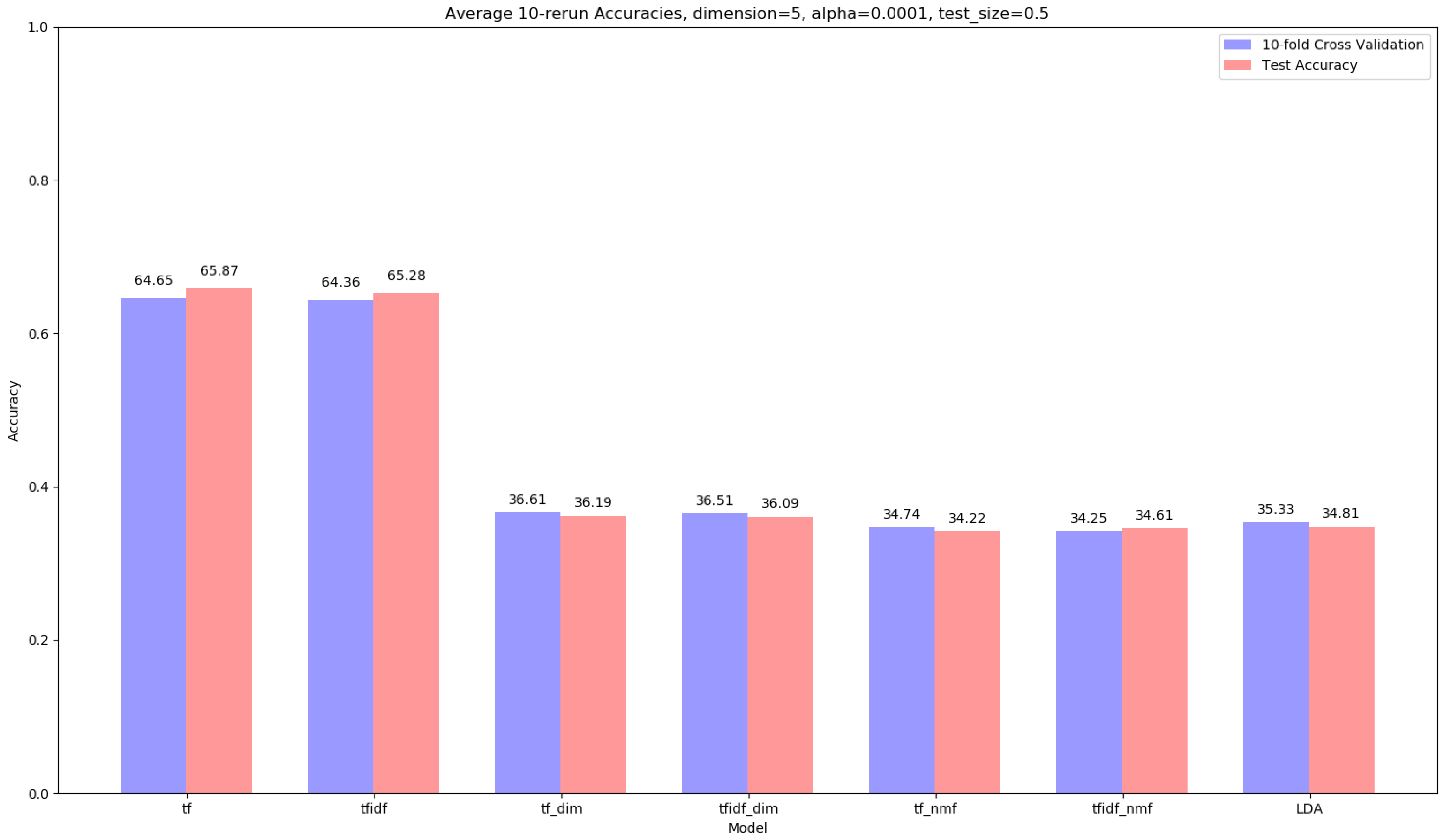
Conclusions

- As α increases, LDA accuracy decreases
- When α is small, LDA works better than other vectorization methods with same dimension

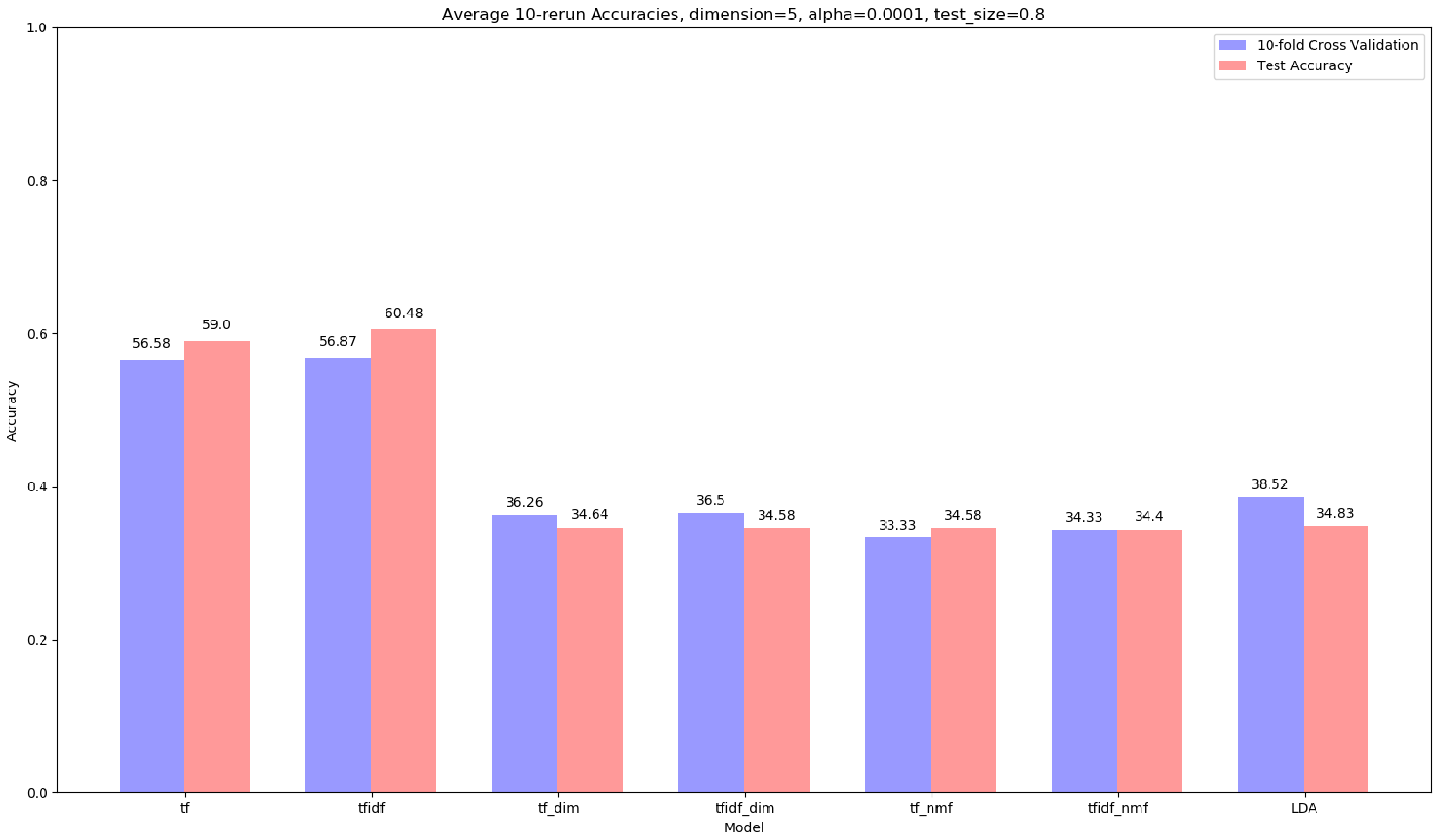
Experiment Result Accuracies with train set ratio=**0.7**, 0.5, 0.2



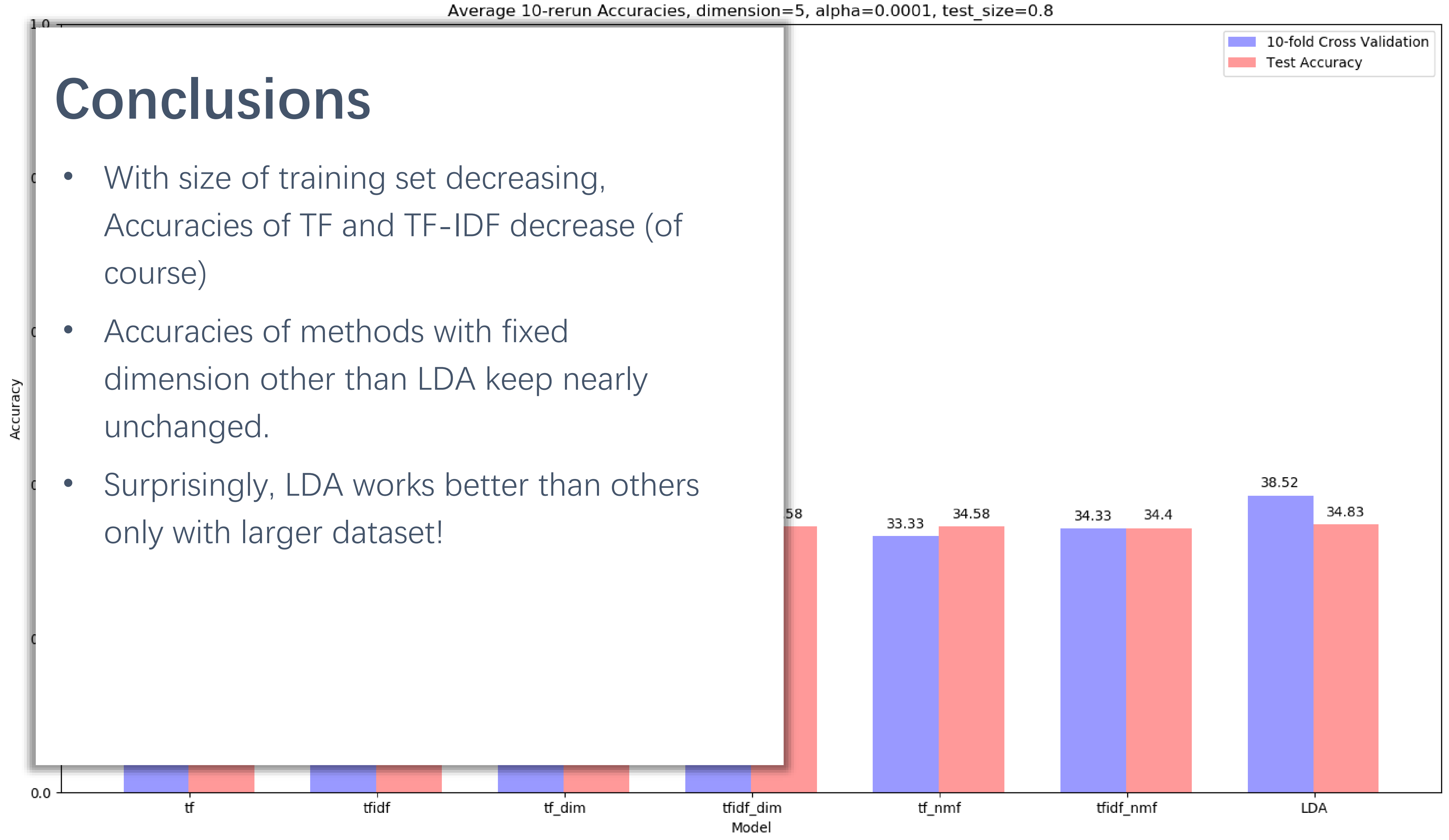
Experiment Result Accuracies with train set ratio=0.7, **0.5**, 0.2



Experiment Result Accuracies with train set ratio=0.7, 0.5, **0.2**



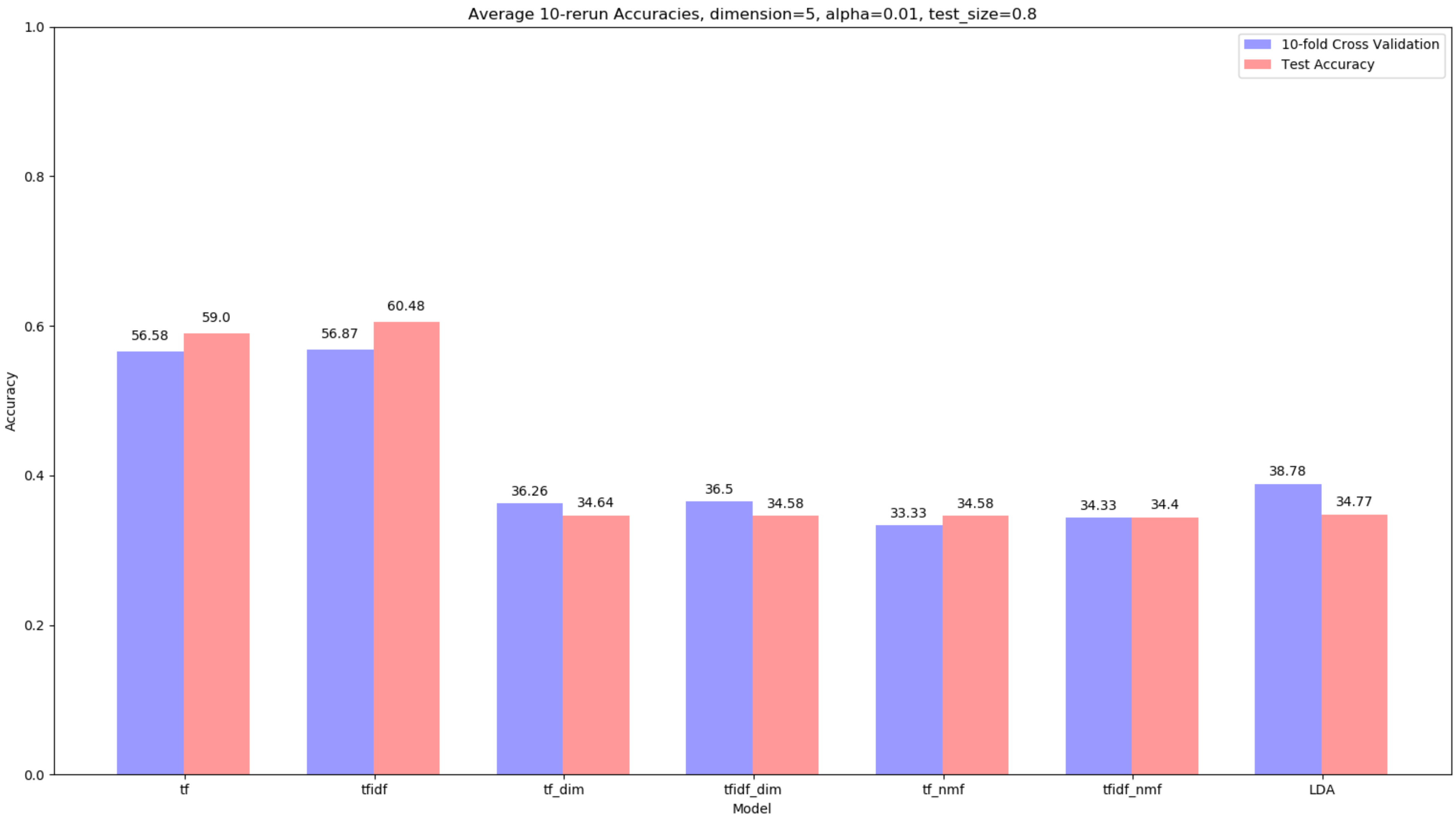
Experiment Result Accuracies with train set ratio=0.7, 0.5, **0.2**



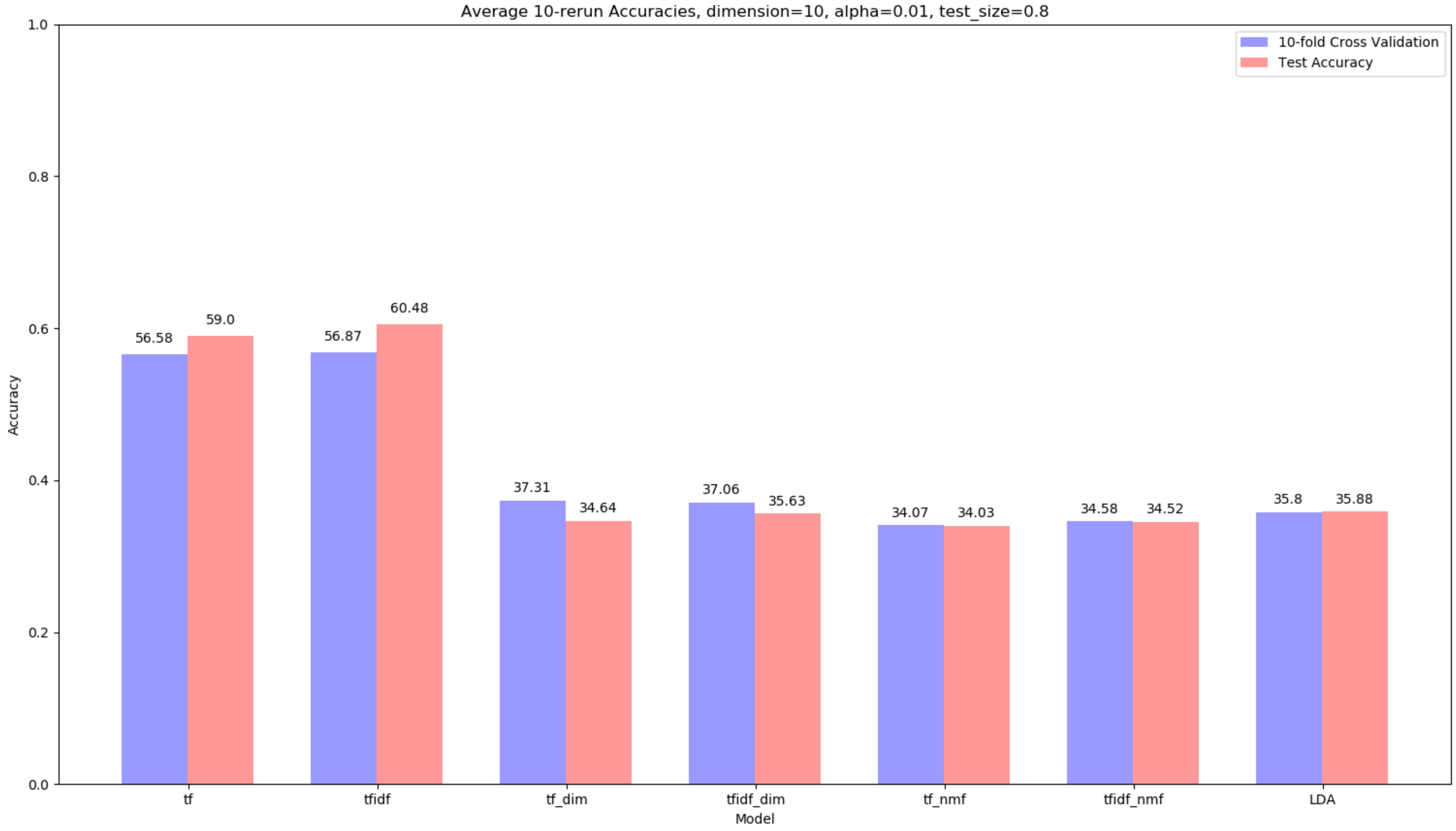
Conclusions

- With size of training set decreasing, Accuracies of TF and TF-IDF decrease (of course)
- Accuracies of methods with fixed dimension other than LDA keep nearly unchanged.
- Surprisingly, LDA works better than others only with larger dataset!

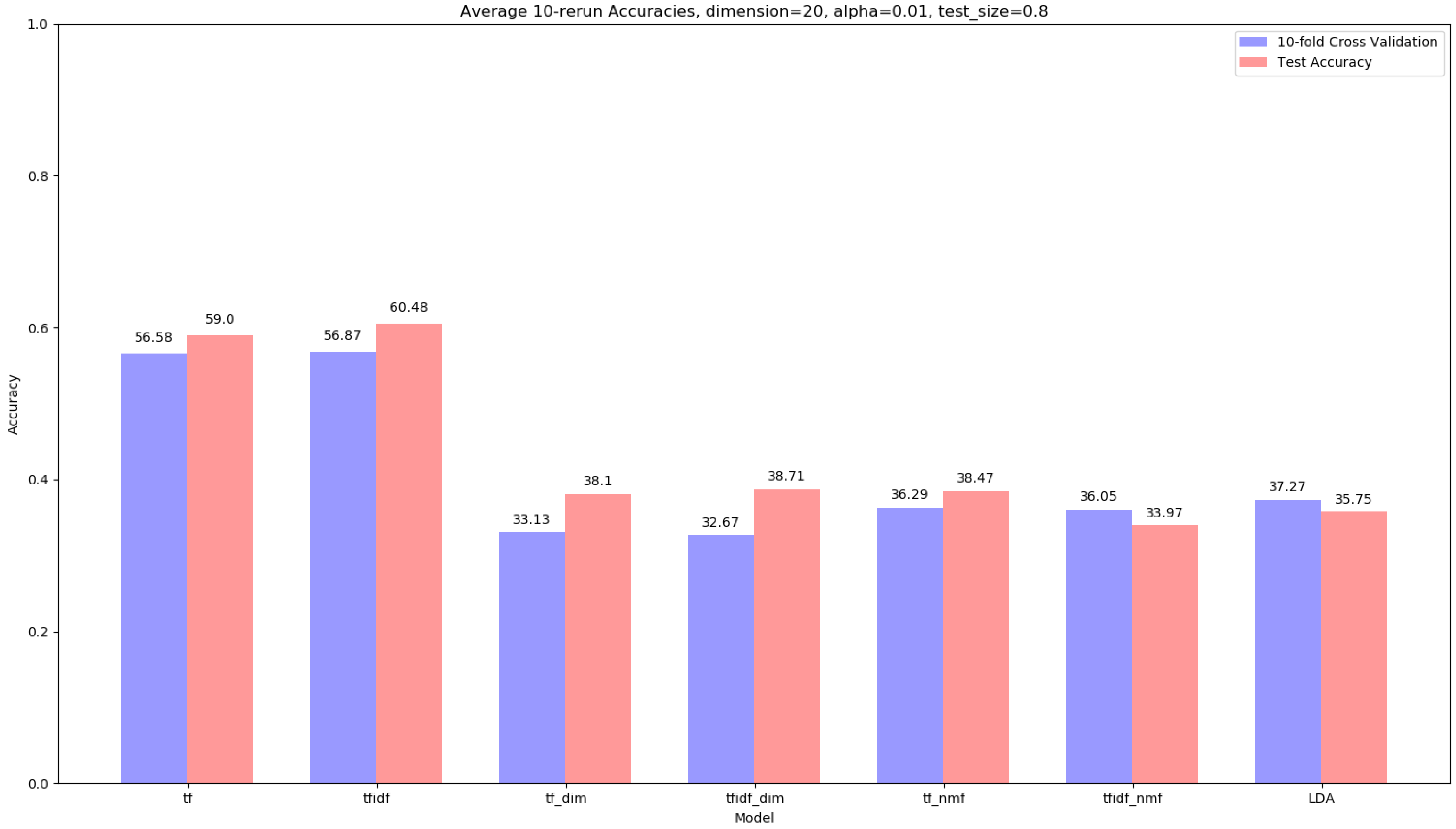
Experiment Result Accuracies with dimension(# of topics)=5, 10, 20



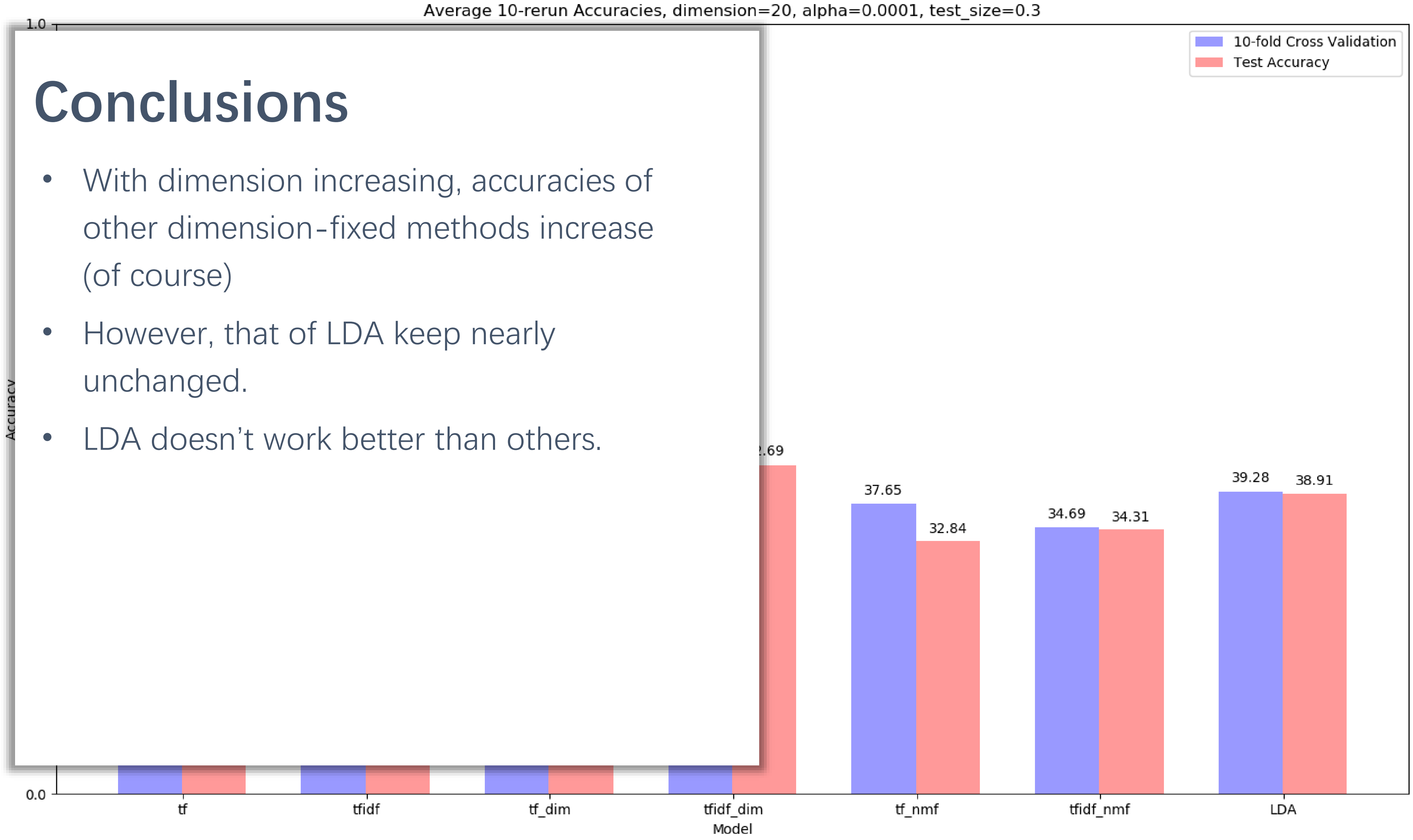
Experiment Result Accuracies with dimension(# of topics)=5, 10, 20



Experiment Result Accuracies with dimension(# of topics)=5, 10, 20



Experiment Result Accuracies with dimension(# of topics)=5, 10, 20



Conclusions

- With dimension increasing, accuracies of other dimension-fixed methods increase (of course)
- However, that of LDA keep nearly unchanged.
- LDA doesn't work better than others.

Experiment Result

Conclusions

- **When should LDA work better (than other document vectorization methods)?**

  - With smaller document-topic-prior (α)?

    Yes!

  - With small dataset?

    No!

  - With lower vector dimension?

    Not better than using other dimension reduction methods.

Questions?