Multi-stage Children Story Speech Synthesis

First Seminar
by

Harikrishna D M
Roll No. 12IT72P07

under the supervision of

Dr. K. Sreenivasa Rao
School of Information Technology
Indian Institute of Technology Kharagpur

August 6, 2015

Overview

Introduction

- Synthesizing expressive speech: Embedding natural expressions into speech, according to the semantics present in the text.
- Story synthesis: Synthesizing story-style speech from the text using text-to-speech (TTS) systems.
- Story synthesis approaches
  - Development of TTS systems using story speech corpus.
  - Rule-based story speech synthesis.
- Application: Audiobooks.

Literature Review: Text Classification

Table: Literature Review in the context of Text Classification

Author	Work	Dataset	Contribution
Joachims (1998) [1]	Text categorization with SVM	Ohsumed (Medical abstracts): 13929 documents, 23 classes	Use of SVM for text classification
Yang et al. (1999) [2]	Examination of text categorization methods	Reuters (News articles): 21578 documents, 90 classes	Controlled study with statistical signicance tests: SVM, KNN, NN, LLSF and NB
Moldovan et al. (2005) [3]	LSA for patent documents	USPTO (Patent documents): 33923 documents, 10 classes	Comparison of VSM and LSA
Sainath et al. (2010) [4]	Sparse representation for text classification	20 Newsgroup (News articles): 20000 documents, 20 classes	Slight improvement in SR method over NB

– Limited to text classification in the domains such as news articles, medical abstracts and patents.

Literature Review: Story-telling Applications

Table: Literature Review in the context of Story-telling Applications

Author	Work	Contribution	Result
Alm et al. (2005) [5]	Perceptions of emotions in expressive storytelling	Analysis of expressive story-telling speech	Semantic and prosodic cues collaborate to express and reinforce emotional content
Lobo et al. (2010) [6]	Fairy tale corpus organization	LSA to represent stories, and recommendation algorithm to define clusters of similar stories	Organized 453 fairy tales from Project Gutenberg
Ceran et al. (2012) [7]	A semantic triplet based story classifier	$< Subject, Verb, Object >$ triplets to identify paragraph as story or not	Better performance with keyword, POS, named entities and semantic triplet features
Iosif et al. (2014) [8]	Multi-step system for children story analysis	Character identification, attribution of quotes and affective analysis of quoted materials	Hybrid approach for children story analysis

– Limited to corpus organization, story analysis and identification.

Literature Review: Indian Languages

Table: Literature Review in the context of Indian Language

Language	Author	Work	Contribution	Result
Punjabi	Nidhi et al. (2012) [9]	Classification of Punjabi news articles	Sports specific ontology, Gazetteer lists	Ontology Based Classification > NB
Marathi	Meera et al. (2014) [10]	Comparison of Marathi text classifiers	Rule based stemmer and Marathi word dictionary	NB > Centroid > Modified KNN > KNN
Kannada	Deepamala et al. (2014) [11]	Kannada Webpage Classification	Sentence boundary detection, stemming, stopword removal	Performance improvement with stemming and stopword removal

Literature Review: Indian Languages (Cont..)

Table: Literature Review in the context of Indian Language

Language	Author	Work	Contribution	Result
Tamil	Rajan et al. (2009) [12]	Tamil document classification	Comparison of VSM and ANN	ANN > VSM
Telugu	Kavi Narayana Murthy (2003) [13]	Telugu News Articles classification	Used NB to classify news articles into Politics, Sports, Business and Cinema	Base system for telugu document classification
Ten Indian Languages	Raghuveer et al. (2007) [14]	Text Categorization in Indian Languages using ML Approaches	Corpus-based machine learning techniques for text categorization	SVM outperformed KNN and NB

– Limited to text classification in the domains such as news articles and web pages.

– None of the works attempted story classification in Indian languages

Scope of present work

- Highly challenging task: Generating an expressive, naturally sounding, story like speech from text using a neutral TTS system.
- Steps in story synthesis

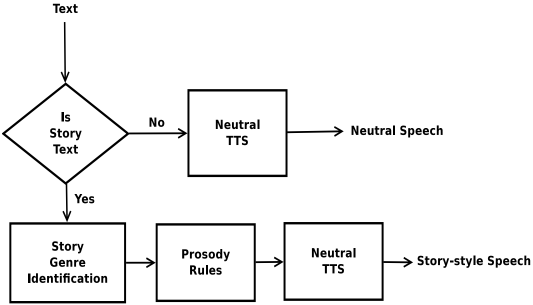

Figure: Overview of steps in story synthesis

Motivation

- Project requirement: *Development of Text-to-Speech systems in Indian languages (Phase - II).*
- Basic objective: To synthesize story style speech from a story text using the neutral text-to-speech (TTS) systems developed in *Phase* – *I* of the project.
- Demo: ▸ Story Text ▸ Neutral TTS Output ▸ Desired Story Style Speech
- Syllable-based unit selection neutral TTS systems developed for six Indian languages in Phase - I of the project [15].
- Each story will be narrated in different style depending on story type.
- Derivation of story specific prosody rules.
- Attempting story classification in view of synthesizing story speech.

Work Done

- Story Classification Framework
- Story Classification using Keyword based Features
- Story Classification using POS Features
- Story Classification using Concatenation of Keyword and POS Features

Story Classification Framework

Story Classification Framework

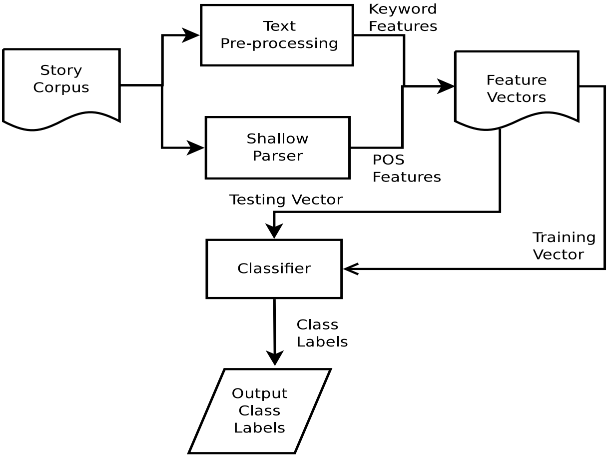

Figure: Flow diagram of Story Classification Framework

Story Corpora

- Hindi and Telugu story corpora: 300 and 150 short stories from Blogs[1], Panchatantra and Akbar-Birbal books.
- Classification of stories into three genres: Fable, folk-tale and legend.
- Definition of story genres
  - Fable: Tale involving animals as an essential character.
  - Folk-tale: Story passed on from one generation to the next.
  - Legend: Story carrying significant meaning or symbolism for the culture.

Table: Details of Hindi and Telugu Story Corpora

Story genre	Hindi		Telugu	
	# Stories	# Words	# Stories	# Words
Fable	100	50344	50	6668
Folk-tale	100	46900	50	6144
Legend	100	35991	50	8540

[1] http://telugubalalu.blogspot.in/

Text Pre-processing and POS Tagging

- Corpus cleaning: Stripping multiple white spaces, removing special symbols and numbers.
- POS tagging and lemmatization: Hindi and Telugu shallow parsers[2] developed by IIIT Hyderabad.
- Lemmatization: Converting word into its root word (base form).
- Stopwords: List of 164 and 138 stopwords.

[2] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

Keyword-based Features

- **"R"** is used for feature extraction.
- **Term Frequency (TF)**: Frequency of terms in a story.
- **Term Frequency Inverse Document Frequency (TFIDF)**: Product of TF and IDF. IDF is calculated as

$$idf(t_i) = log \frac{N}{n_i}$$

where N is the total number of stories and n_i is the number of stories in the corpus that contains word t_i.

Linguistic-based features

- POS: Category of words having similar grammatical property.
- POS tags: Noun (NN), Proper Noun (NNP), Spatial and Temporal Nouns (NST), Pronoun (PRP), Finite Verb (VM), Auxiliary Verb (VAUX), Post Position (PSP), Particles (RP), Adjective (JJ) and Quantifiers (QF).
- Relevance of the POS tags with respect to Indian languages are explained in shallow parser manual[3].
- **POS Density (PD)**: For each story, PD is calculated as

$$PD = \sum_{p \in P} \frac{count(p)}{Total\ words\ in\ story}$$

where P = NN, VM, PSP, PRP, NNP, NST, JJ and QF.

[3] http://ltrc.iiit.ac.in/tr031/posguidelines.pdf

Classifiers

- Combinations of features: PD, TF, TFIDF, TF + PD and TFIDF + PD.
- Three promising machine learning classifiers: Naive Bayes (NB), K-Nearest Neighbour (KNN), Support Vector Machine (SVM).
- 10-fold cross validation, nine nearest neighbours (k=9), linear kernel for SVM.
- Implementation of classifiers: WEKA combined with LibSVM package.

Evaluation Measures

$$Precision\ (P) = \frac{No.\ of\ stories\ correctly\ classified\ as\ class\ ``x"}{No.\ of\ stories\ classified\ as\ class\ ``x"}$$

$$Recall\ (R) = \frac{No.\ of\ stories\ correctly\ classified\ as\ class\ ``x"}{Actual\ No.\ of\ stories\ of\ class\ ``x"}$$

$$F - measure\ (F) = \frac{2 \times P \times R}{(P + R)}$$

$$Accuracy = \frac{No.\ of\ stories\ correctly\ classified}{Total\ No.\ of\ stories}$$

$$Macro\ F1 = \frac{\sum_{i \in C} F_i}{\mid C \mid}$$

where C is the set of predefined classes and F_i is the F-measure for the i^{th} class in C.

– Statistical significance test: McNemar's test.

Story Classification using Keyword based Features

Story Classification using Keyword based Features

- Document-term matrix (DTM): Each row represents a story and each column represents a term in the collection.
- DTM: Huge feature size and highly sparse.
- Better performance can be achieved by optimal representation of features.
- Feature reduction techniques: Sparse Term Removal, Latent Semantic Analysis (LSA).
- Sparseness factors: 0.7, 0.75, 0.8, 0.85, 0.9 and 0.95.
- LSA: Values of k for Hindi and Telugu respectively are $\{25, 50, 75, 100, 125, 150\}$ and $\{15, 30, 45, 60, 75, 90\}$.

Results of Story Classification using Keyword based Features

Table: Macro F1 measure for story classification using feature reduction techniques for Hindi

Classifiers	Full Story	Dimension Reduction Techniques											
		Sparseness Factor						LSA					
		0.7	0.75	0.8	0.85	0.9	0.95	25	50	75	100	125	150
	300×6608	300×78	300×104	300×143	300×182	300×366	300×681	300×25	300×50	300×75	300×100	300×125	300×150
NB	0.71	0.81	0.83	0.84	0.86	**0.89**	0.84	0.4	0.4	0.41	0.41	**0.43**	0.42
KNN	0.61	0.71	0.73	0.74	0.75	**0.77**	0.73	0.62	0.63	0.63	0.67	**0.68**	0.65
SVM	0.62	0.79	0.82	0.85	0.86	**0.91**	0.82	0.32	0.37	0.41	0.46	**0.48**	0.47

Table: Macro F1 measure for story classification using feature reduction techniques for Telugu

Classifiers	Full Story	Dimension Reduction Techniques											
		Sparseness Factor						LSA					
		0.7	0.75	0.8	0.85	0.9	0.95	15	30	45	60	75	90
	150×4539	150×17	150×29	150×49	150×88	150×232	150×582	150×15	150×30	150×45	150×60	150×75	150×90
NB	0.76	0.78	0.8	0.81	0.83	**0.86**	0.8	0.64	0.66	**0.67**	0.61	0.56	0.54
KNN	0.46	0.68	0.7	0.72	0.73	**0.75**	0.71	0.63	0.65	**0.71**	0.63	0.58	0.46
SVM	0.81	0.84	0.85	0.87	0.89	**0.94**	0.87	0.44	0.51	**0.58**	0.56	0.55	0.52

Analysis of Results of Story Classification using Keyword based Features

- Increasing the sparseness factor, the most frequently repeated terms in story corpora are included in DTM.
- Increasing the sparseness factor beyond a threshold can add noisy terms, which do not contribute for identifying the story genre and thus decreases the performance.
- LSA failed to capture the behaviour of implicit higher-order structure by lower dimensional document-term matrix.
- Conclusion: Sparseness factor of **0.9** assures a good performance.

Story Classification using POS Features

Distribution of POS tags

– Motivation for selecting POS: More named entities in stories, POS such as nouns, adjectives, quantifiers and verbs are useful feature for distinguishing between story genres.

Table: POS distribution across story genres

POS Tags	Hindi			Telugu		
	Fable	Folk-tale	Legend	Fable	Folk-tale	Legend
NN	10975	9985	7277	2539	2386	2957
VM	9298	8439	6098	1919	1730	2377
PSP	6788	6249	4898	104	110	131
PRP	5286	4910	3761	615	557	769
VAUX	4278	3735	2817	40	38	48
JJ	1691	1698	1420	264	217	238
NNP	1534	1497	1554	22	152	516
RP	1456	1353	1011	45	38	86
NST	1035	764	584	275	178	283
QF	635	530	503	61	40	75

POS Tag Sets

- Unclear that which class of POS tags like Nouns, Verbs, Adjectives, Quantifiers, Particles or Post position are necessary for recognition of story genres.
- Different combination of POS tags: Investigation of the effect of linguistic information on story classification.

Table: Different sets of POS tags

Set	POS Tags
Set 1	$\{NN, NNP, NST, PRP, JJ, QF, VM, VAUX, PSP, RP\}$
Set 2	$\{NN, NNP, NST, PRP, JJ, QF\}$
Set 3	$\{NN, NNP, NST, PRP, VM, VAUX\}$
Set 4	$\{NN, NNP, NST, PRP, PSP, RP\}$
Set 5	$\{NN, NNP, NST, PRP\}$
Set 6	$\{JJ, QF, VM, VAUX\}$

Performance Measures for Different POS Tag Sets

Table: Macro F1 measures for different sets of POS tags

Set	Hindi			Telugu		
	NB	KNN	SVM	NB	KNN	SVM
Set 1	0.48	0.4	0.45	0.55	0.47	0.56
Set 2	**0.49**	**0.43**	**0.5**	**0.56**	**0.55**	**0.58**
Set 3	0.48	0.4	0.48	0.55	0.51	0.57
Set 4	0.48	0.38	0.47	0.54	0.52	0.56
Set 5	0.45	0.4	0.46	0.53	0.51	0.56
Set 6	0.42	0.33	0.39	0.38	0.38	0.36

- POS tags are similar across stories, hence they cannot be as contributing as keyword based features.
- Conclusion: Nouns, adjectives and quantifiers have contributed more to the story classification.

Story Classification using Concatenation of Keyword and POS Features

Results of Story Classification using Concatenation of Keyword and POS Features

Table: Performance measures for story classificaiton using concatenation of keyword and POS features

Story Genre	Features	Hindi									Telugu								
		NB			KNN			SVM			NB			KNN			SVM		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Fable	PD	0.46	0.65	0.54	0.47	0.70	0.57	0.46	0.48	0.48	0.56	0.62	0.59	0.48	0.72	0.58	0.59	0.96	0.73
	TF	0.93	0.88	0.9	0.89	0.59	0.71	0.94	0.91	0.92	0.95	0.8	0.87	0.68	0.7	0.69	0.94	0.94	0.94
	TF + PD	0.93	0.90	**0.91**	0.89	0.68	**0.77**	0.95	0.93	**0.94**	0.98	0.82	**0.89**	0.78	0.76	**0.77**	0.98	0.96	**0.97**
	TFIDF	0.89	0.44	0.59	0.86	0.56	0.68	0.92	0.9	0.91	0.86	0.74	0.8	0.64	0.64	0.64	0.92	0.92	0.92
	TFIDF + PD	0.9	0.75	**0.81**	0.88	0.66	**0.75**	0.94	0.92	**0.93**	0.93	0.78	**0.85**	0.72	0.68	**0.7**	0.94	0.92	**0.93**
Folk-tale	PD	0.63	0.35	0.45	0.38	0.31	0.34	0.52	0.41	0.46	0.46	0.72	0.56	0.55	0.30	0.39	0.58	0.22	0.32
	TF	0.87	0.87	0.87	0.66	0.84	0.74	0.96	0.9	0.93	0.75	0.92	0.83	0.86	0.76	0.8	0.96	0.92	0.94
	TF + PD	0.87	0.90	**0.89**	0.75	0.86	**0.8**	0.97	0.92	**0.94**	0.76	0.94	**0.84**	0.8	0.82	**0.81**	0.98	0.94	**0.96**
	TFIDF	0.76	0.76	0.76	0.65	0.82	0.73	0.94	0.89	0.91	0.74	0.84	0.78	0.78	0.72	0.75	0.94	0.9	0.92
	TFIDF + PD	0.82	0.8	**0.81**	0.7	0.83	**0.76**	0.94	0.9	**0.92**	0.79	0.86	**0.82**	0.76	0.78	**0.77**	0.96	0.92	**0.94**
Legend	PD	0.59	0.39	0.47	0.49	0.34	0.40	0.54	0.54	0.54	0.87	0.40	0.55	0.75	0.60	0.67	0.72	0.62	0.67
	TF	0.87	0.93	0.9	0.85	0.9	0.87	0.85	0.94	0.89	0.91	0.86	0.88	0.72	0.8	0.76	0.92	0.96	0.93
	TF + PD	0.96	0.96	**0.96**	0.84	0.92	**0.88**	0.9	0.96	**0.93**	0.96	0.88	**0.92**	0.84	0.84	**0.84**	0.92	0.98	**0.95**
	TFIDF	0.64	0.96	0.77	0.82	0.9	0.86	0.86	0.92	0.88	0.82	0.82	0.82	0.68	0.74	0.71	0.9	0.94	0.91
	TFIDF + PD	0.74	0.88	**0.8**	0.84	0.91	**0.87**	0.87	0.93	**0.9**	0.81	0.88	**0.84**	0.77	0.8	**0.78**	0.9	0.96	**0.93**

Story Classification Accuracy using Concatenation of Keyword and POS Features

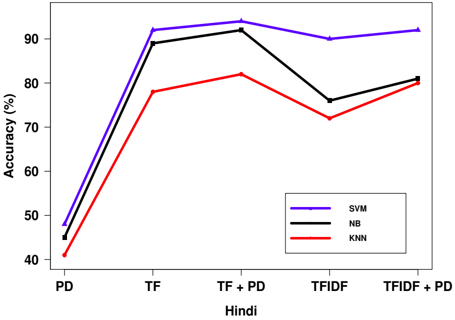

Figure: Story classification accuracy using concatenation of keyword and POS features

McNemar's Significance Test Results for Different Combinations of Features

Table: Statistical significance test results for different combination of features

Classifier	Hindi		Telugu	
	TF + PD vs TF	TFIDF + PD vs TFIDF	TF + PD vs TF	TFIDF + PD vs TFIDF
NB	$>$	$\sim$	$>$	$\sim$
KNN	$\sim$	$\sim$	$\sim$	$\sim$
SVM	$>$	$>$	$>$	$>$

" $>$ " means $0.01 < P\text{-}value \leq 0.05$, which is statistically significant
" $\sim$ " means $P\text{-}value > 0.05$, which is not statistically significant

Demo

McNemar's Significance Test Results for Cross-classifier Performance

Table: Statistical significance test results for cross-classifier performance

Classifier A	Classifier B	Hindi					Telugu				
		PD	TF	TF + PD	TFIDF	TFIDF + PD	PD	TF	TF + PD	TFIDF	TFIDF + PD
NB	KNN	$\sim$	$\gg$	$\gg$	$>$	$\sim$	$\sim$	$\gg$	$\gg$	$\gg$	$\gg$
SVM	KNN	$\sim$	$\gg$	$\gg$	$\gg$	$\gg$	$\sim$	$\gg$	$\gg$	$\gg$	$\gg$
SVM	NB	$\sim$	$\sim$	$\sim$	$\gg$	$\gg$	$\sim$	$\gg$	$>$	$\gg$	$>$

" $\gg$ " means *P-value* ≤ 0.01, which is extremely statistically significant
" $>$ " means $0.01 <$ *P-value* ≤ 0.05, which is statistically significant
" $\sim$ " means *P-value* > 0.05, which is not statistically significant

Demo

Analysis of Results of Story Classification using Concatenation of Keyword and POS Features

- NB is a probabilistic learning method. It is based on Bayes theorem and the story genre will be assigned to the class having maximum a posteriori probability.
- The poor performance of KNN can be due to the noisy terms in the DTM.
- SVM has better performance because it is resilient to noise.

Summary and Conclusions

- Contributions
  - Developed story corpora for Hindi and Telugu.
  - Story Classification using Concatenation of Keyword and POS Features.
- Conclusions
  - In case of feature reduction techniques, sparseness factor of **0.9** gave the highest performance.
  - Using linguistic information boosts the performance of story classification significantly.
  - POS tag set consisting of nouns, adjectives and quantifiers have the highest accuracy and are important for story classification.
  - In most of the cases, the highest performance is achieved by TF + PD features and SVM models outperformed the other models in terms of classification accuracy.

Future Work

- *Story classification using partial story information:* Exploring story classification by dividing stories into parts based on story semantics.
- *Emotion prediction from story text:* Exploring Keyword, POS and story specific features for predicting emotion from story text.
- *Deriving prosody rules:* Deriving prosody rules (modification factors) specific to emotions and story genres.
- *Synthesis of story speech using mark-up language:* Story-specific prosody rules can be effectively incorporated using SABLE mark-up language. The quality and naturalness of the synthesized story speech can be evaluated using subjective tests.

Publications

- **Conference**
  - Harikrishna D M and K. Sreenivasa Rao, "*Classification of Children Stories in Hindi Using Keywords and POS Density,*" in *International Conference on Computer Communication and Control (IC4)*, Indore, 2015.

ACKNOWLEDGMENTS

We are thankful to the **Department of Information Technology, Govt. of India** for supporting the research work, **“Development of Text-to-Speech synthesis for Indian Languages Phase II”**

References I

[1] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[2] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 42–49, 1999.

[3] A. Moldovan, R. I. Bot, and G. Wanka, "Latent semantic indexing for patent documents," *International Journal of Applied Mathematics and Computer Science*, vol. 15, pp. 551–591, 2005.

[4] T. N. Sainath, S. Maskey, D. Kanevsky, B. Ramabhadran, D. Nahamoo, and J. Hirschberg, "Sparse representations for text categorization," in *INTERSPEECH*, pp. 2266–2269, 2010.

References II

[5] C. O. Alm and R. Sproat, “Perceptions of emotions in expressive storytelling,” in *INTERSPEECH*, pp. 533–536, 2005.

[6] P. V. Lobo and D. M. De Matos, “Fairy tale corpus organization using latent semantic mapping and an item-to-item top-n recommendation algorithm,” in *Language Resources and Evaluation Conference (LREC)*, 2010.

[7] B. Ceran, R. Karad, A. Mandvekar, S. R. Corman, and H. Davulcu, “A semantic triplet based story classifier,” in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2012.

[8] E. Iosif and T. Mishra, “From speaker identification to affective analysis: A multi-step system for analyzing children stories,” *European Chapter of the ACL (EACL)*, pp. 40–49, 2014.

[9] Nidhi and V. Gupta, "Domain based classification of Punjabi text documents using ontology and hybrid based approach," in *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing, COLING*, 2012.

[10] M. Patil and P. Game, "Comparison of Marathi Text Classifiers," *Association of Computer Electronics and Electrical Engineers*, vol. 4, 2014.

[11] N. Deepamala and P. R. Kumar, "Text Classification of Kannada webpages using various pre-processing agents," in *Recent Advances in Intelligent Informatics*. Springer, 2014.

[12] K. Rajan, V. Ramalingam, M. Ganesan, S. Palanivel, and B. Palaniappan, "Automatic classification of Tamil documents using vector space model and artificial neural network," *Expert Systems with Applications*, vol. 36, pp. 10914–10918, 2009.

References IV

[13] K. N. Murthy, "Automatic categorization of Telugu news articles," *Department of Computer and Information Sciences*, 2003.

[14] K. Raghuveer and K. N. Murthy, "Text categorization in Indian languages using machine learning approaches," in *Indian International Conference on Artificial Intelligence*, 2007.

[15] H. A. Patil, T. B. Patel, N. J. Shah, H. B. Sailor, R. Krishnan, G. Kasthuri, T. Nagarajan, L. Christina, N. Kumar, V. Raghavendra *et al.*, "A syllable-based framework for unit selection synthesis in 13 Indian languages," in *Oriental COCOSDA held jointly with Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*. IEEE, pp. 1–8, 2013.

Thank You

Backup Slides

McNemar's significance test

- Contingency table

η_{00} : Number of examples misclassified by both classifiers C_A and C_B	η_{01} : Number of examples misclassified by classifier C_A but not by C_B
η_{10} : Number of examples misclassified by classifier C_B but not by C_A	η_{11} : Number of examples misclassified by neither classifiers C_A nor C_B

- The statistic χ is defined as

$$\chi = \frac{(\mid \eta_{01} - \eta_{10} \mid - 1)^2}{\eta_{01} + \eta_{10}}$$

Sparse Term Removal Example

```
Story_1.txt - Story one text example
Story_2.txt - Story two text example
Story_3.txt - Story three text example
Story_4.txt - Story four text example
Story_5.txt - Story five text example
Story_6.txt - Story six text example
Story_7.txt - Story seven text
Story_8.txt - Story eight text
Story_9.txt - Story nine
Story_10.txt - Story ten
```

Figure: Story text

Document Term Matrix

```
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
              Terms
Docs           eight example five four nine one seven six story ten text three two
  Story_10.txt     0       0    0    0    0   0     0   0     1   1    0     0   0
  Story_1.txt      0       1    0    0    0   1     0   0     1   0    1     0   0
  Story_2.txt      0       1    0    0    0   0     0   0     1   0    1     0   1
  Story_3.txt      0       1    0    0    0   0     0   0     1   0    1     1   0
  Story_4.txt      0       1    0    1    0   0     0   0     1   0    1     0   0
  Story_5.txt      0       1    1    0    0   0     0   0     1   0    1     0   0
  Story_6.txt      0       1    0    0    0   0     0   1     1   0    1     0   0
  Story_7.txt      0       0    0    0    0   0     1   0     1   0    1     0   0
  Story_8.txt      1       0    0    0    0   0     0   0     1   0    1     0   0
  Story_9.txt      0       0    0    0    1   0     0   0     1   0    0     0   0
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
> |
```

Figure: Document Term Matrix

Sparse Term Removal

- Sparseness factor = 0.1
- Remove terms which have greater than 10% percentage of empty elements or get terms which exists in 90% of stories.

```
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
              Terms
Docs            story
  Story_10.txt      1
  Story_1.txt       1
  Story_2.txt       1
  Story_3.txt       1
  Story_4.txt       1
  Story_5.txt       1
  Story_6.txt       1
  Story_7.txt       1
  Story_8.txt       1
  Story_9.txt       1
<<DocumentTermMatrix (documents: 10, terms: 1)>>
Non-/sparse entries: 10/0
Sparsity           : 0%
Maximal term length: 5
Weighting          : term frequency (tf)
> 
```

Figure: With Sparseness factor of 0.1

Sparse Term Removal (Cont...)

- Sparseness factor = 0.2
- Remove terms which have greater than 20% percentage of empty elements or get terms which exists in 80% of stories.

```
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
              Terms
Docs           story text
  Story_10.txt     1    0
  Story_1.txt      1    1
  Story_2.txt      1    1
  Story_3.txt      1    1
  Story_4.txt      1    1
  Story_5.txt      1    1
  Story_6.txt      1    1
  Story_7.txt      1    1
  Story_8.txt      1    1
  Story_9.txt      1    0
<<DocumentTermMatrix (documents: 10, terms: 2)>>
Non-/sparse entries: 18/2
Sparsity           : 10%
Maximal term length: 5
Weighting          : term frequency (tf)
>
```

Figure: With Sparseness factor of 0.2

- Sparseness factor $= 0.4$
- Remove terms which have greater than 40% percentage of empty elements or get terms which exists in 60% of stories.

```
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
              Terms
Docs            example story text
  Story_10.txt        0     1    0
  Story_1.txt         1     1    1
  Story_2.txt         1     1    1
  Story_3.txt         1     1    1
  Story_4.txt         1     1    1
  Story_5.txt         1     1    1
  Story_6.txt         1     1    1
  Story_7.txt         0     1    1
  Story_8.txt         0     1    1
  Story_9.txt         0     1    0
<<DocumentTermMatrix (documents: 10, terms: 3)>>
Non-/sparse entries: 24/6
Sparsity           : 20%
Maximal term length: 7
Weighting          : term frequency (tf)
> 
```

Figure: With Sparseness factor of 0.4

- Sparseness factor = 0.9.
- Remove terms which have greater than 90% percentage of empty elements or get terms which exists in 10% of stories.
- Same as without sparse term removal

```
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
              Terms
Docs           eight example five four nine one seven six story ten text three two
  Story_10.txt     0       0    0    0    0   0     0   0     1   1    0     0   0
  Story_1.txt      0       1    0    0    0   1     0   0     1   0    1     0   0
  Story_2.txt      0       1    0    0    0   0     0   0     1   0    1     0   1
  Story_3.txt      0       1    0    0    0   0     0   0     1   0    1     1   0
  Story_4.txt      0       1    0    1    0   0     0   0     1   0    1     0   0
  Story_5.txt      0       1    1    0    0   0     0   0     1   0    1     0   0
  Story_6.txt      0       1    0    0    0   0     0   1     1   0    1     0   0
  Story_7.txt      0       0    0    0    0   0     1   0     1   0    1     0   0
  Story_8.txt      1       0    0    0    0   0     0   0     1   0    1     0   0
  Story_9.txt      0       0    0    0    1   0     0   0     1   0    0     0   0
<<DocumentTermMatrix (documents: 10, terms: 13)>>
Non-/sparse entries: 34/96
Sparsity           : 74%
Maximal term length: 7
Weighting          : term frequency (tf)
> 
```

Figure: With Sparseness factor of 0.9

Latent Semantic Analysis

- Basic Idea: Let C be a DTM $(M \times N)$ with non-negative real valued entries and $m = min(M, N)$. C can be decomposed into a set of k orthogonal matrices whose linear combination is a good approximation of initial matrix C.
- Formal definition: C can be decomposed as, $C = USV^T$; where matrices $U(M \times m)$ and $V(N \times m)$ are orthonormal matrices $(U^T U = I_m$ and $V^T V = I_m)$ whose columns define left and right singular vectors respectively and S is a $m \times m$ diagonal matrix of singular values of C decreasingly ordered along its diagonal.
- Retain only the k greatest singular values in S ,then the product of resulting matrices S_k, U_k and V_k is the best approximation of original C by a matrix of rank k

$$C \simeq C_k = U_k S_k V_k^T$$

where C_k is the approximation of original document-term matrix C, S_k is a diagonal matrix consisting of largest k values.

LSA Example

- Source: Introduction to Information Retrieval (Manning et al., 2008)
- Consider a term-document matrix C

Example 18.4: Consider the term-document matrix C =

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Figure: Term document matrix

- Matrix U

	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
voyage	−0.70	0.35	0.15	−0.58	0.16
trip	−0.26	0.65	−0.41	0.58	−0.09

Figure: SVD term matrix

- Matrix S

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

Figure: Singular Values matrix

- Matrix V^T

	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

Figure: SVD document matrix

LSA Example (Cont...)

- When $k = 2$, Matrix S

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

Figure: Singular Values matrix for k = 2

- Matrix C_2

	d_1	d_2	d_3	d_4	d_5	d_6
1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
2	−0.46	−0.84	−0.30	1.00	0.35	0.65
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Figure: Term document matrix for k = 2

LSA Example (Cont...)

– Term document matrix C reduced to two dimensions.

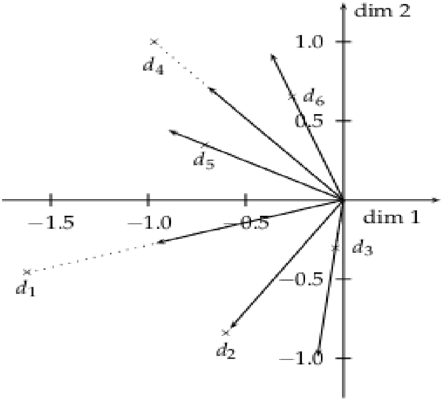

Figure: Term document matrix reduced to two dimensions

LSA Result Analysis

- LSA captures most of underlying structure in association of terms and documents.
- Since $k \ll terms$, it is expected that terms which occur in similar stories will be near each other in k dimensional space even though if they never co-occur in same stories.
- Some stories which do not share any words in common, may however be near in $k - dimensional$ space.

Confusion Matrix

– Confusion matrix for various classifiers using TF + PD features for Hindi. (A) indicates actual and (P) indicates predicted.

Table: Confusion matrix for NB

	Fable (P)	Folk-tale (P)	Legend (P)
Fable (A)	88	8	4
Folk-tale (A)	9	89	2
Legend (A)	5	5	90

Table: Confusion matrix for KNN

	Fable (P)	Folk-tale (P)	Legend (P)
Fable (A)	68	25	7
Folk-tale (A)	13	80	7
Legend (A)	6	4	90

Table: Confusion matrix for SVM

	Fable (P)	Folk-tale (P)	Legend (P)
Fable (A)	92	2	6
Folk-tale (A)	2	90	8
Legend (A)	3	4	93