

# Abstractive Text Summarization

---

Isaac Koak

November 19, 2016

University of Minnesota, Morris

# Disappointing Wiki :(

## Hans Stubb

From Wikipedia, the free encyclopedia

**Hans "Hennes" Stubb** (8 October 1906 - 19 March 1973) was a German footballer.

He played in defense for Eintracht Frankfurt from 1928 to 1944. He also played 10 times for Germany, scoring one goal.

He shot his only international against Hungary on a wet pitch from 60 metres.

Hans Stubb is listed as an honoured captain at Eintracht Frankfurt.

### External links [[edit](#)]

- [Hans Stubb at eintracht-archiv.de](#)



*This biographical article related to association football in Germany, about a midfielder born in the 1900s, is a stub. You can help Wikipedia by expanding it.*

### Hennes Stubb

Personal information		
<b>Full name</b>	Hans Stubb	
<b>Date of birth</b>	October 8, 1906	
<b>Place of birth</b>	Germany	
<b>Date of death</b>	March 19, 1973 (aged 66)	
<b>Playing position</b>	Winger	
Youth career		
1920-1925	Frankfurter FC Germania 1894	
Senior career*		
Years	Team	Apps (Gls)
1925-1928	<b>SpVgg Ostend 07 Frankfurt</b>	
1928-1944	Eintracht Frankfurt	139 (9)
National team		
1930-1934	Germany	10 (1)
		* Senior club appearances and goals counted for the domestic league only.

[clicked red link] ↓

## SpVgg Ostend 07 Frankfurt

From Wikipedia, the free encyclopedia

**Wikipedia does not have an article with this exact name.** Please [search for SpVgg Ostend 07 Frankfurt](#) in Wikipedia to check for alternative titles or spellings.

- Log in or create an account to start the **SpVgg Ostend 07 Frankfurt** article, alternatively use the Article Wizard, or add a request for it.
- Search for "SpVgg Ostend 07 Frankfurt" in existing articles.
- Look for pages within Wikipedia that link to this title.

#### Other reasons this message may be displayed:

- If a page was recently created here, it may not be visible yet because of a delay in updating the database; wait a few minutes or try the [purge](#) function.
- Titles on Wikipedia are **case sensitive** except for the first character; please check [alternative capitalizations](#) and consider adding a [redirect](#) here to the correct title.
- If the page has been deleted, check the [deletion log](#), and see [Why was the page I created deleted?](#)

Look for **SpVgg Ostend 07 Frankfurt** on

one of Wikipedia's sister projects:

- Wiktionary (free dictionary)
- Wikibooks (free textbooks)
- Wikiquote (quotations)
- Wikisource (free library)
- Wikiversity (free learning resources)
- Commons (images and media)
- Wikivoyage (free travel guide)
- Wikinews (free news source)
- Wikidata (free linked database)

## 1. Introduction

Automatic Summarization and Motivation

## 2. Background

N-gram

Document Vector Space and Cosine Similarity

## 3. Application

WikiWrite

## 4. Results

Summary Evaluation Metrics

WikiWrite Results

# Introduction

---

Automatic Summarization and  
Motivation

# Automatic Summarization

Goal: produce a concise summary using a computer program that retains the most important points of the original document

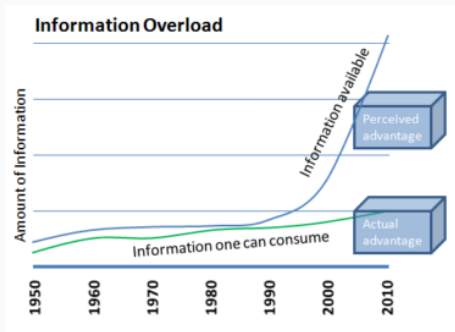
Two categories:

- **Extractive** - concatenate words, phrases, or sentences in the original document to form summary
- **Abstractive** - attempts a deeper analysis of the text and summarize using own words

## Human-written Revision Operations: Hongyan Jing, 2002

Operation	Extractive	Abstractive
Sentence Reduction	✓	✓
Sentence Combination	✓	✓
Syntactic Transformation	✓	✓
Lexical Paraphrasing		✓
Generalization or Specification		✓
Reordering	✓	✓

# Motivation: Why Abstractive Text Summarization?



**Information overload** - *difficulty a person can have understanding an issue and making decisions that can be caused by the presence of too much information.* - Wikipedia

- Extractive methods stagnating
- Greater range of application
- Future looks exciting

# Background

---

N-gram



**n-gram** - a continuous sequence of n words

- Example - “the cat is black”
  - unigrams - the, cat, is, black
  - bigrams - the cat, cat is, is black
  - trigrams - the cat is, cat is black
  - 1-skip-bigram - the is, cat black

# Term Frequency (TF)

- TF - how many times a term appears in a document.

Term	Count
the	3
cat	1
is	3
black	1

**Table 1:** Document 1

$$TF(t, d) = \frac{\text{number of times a term } (t) \text{ appears in a document } (d)}{\text{total number of terms in the document } (d)}$$

$$TF(\text{"the"}) = \frac{3}{8} = 0.38$$

$$TF(\text{"cat"}) = \frac{1}{8} = 0.13$$

# Term Frequency (TF): Stop Words

**stop word** - common words that carry little value

- English: a, an, the, is, on, that
- Biology: cell
- Computer Science: algorithm

# Term Frequency-Inverse Document Frequency (TF-IDF)

$$TF(t, d) = \frac{\text{number of times a term } (t) \text{ appears in a document } (d)}{\text{total number of terms in the document } (d)}$$

$$IDF(t, D) = \log \left( \frac{\text{total number of documents } (D)}{\text{number of documents with term } t} \right)$$

- IDF - measure how informative a term is in a document.
- TF-IDF - how important a term is to a document in a collection of documents.

$$TFIDF = TF * IDF$$

## TF-IDF: Example

Term	Count
the	3
cat	1
is	3
black	1

**Table 2:** Document 1

Term	Count
the	3
dog	1
is	3
black	1

**Table 3:** Document 2

$$TF(\text{"the"}) = \frac{3}{8} = 0.38$$

$$IDF(\text{"the"}, D) = \log\left(\frac{2}{2}\right) = 0$$

$$TF(\text{"dog"}, d_2) = \frac{1}{8} = 0.13$$

$$IDF(\text{"dog"}, D) = \log\left(\frac{2}{1}\right) = 0.30$$

$$TFIDF(\text{"dog"}, d_2, D) = 0.13 * 0.30 = 0.04$$

$$TFIDF(\text{"the"}, d_2, D) = 0.38 * 0 = 0.00$$

# Background

---

Document Vector Space and Cosine Similarity

# Cosine Similarity: Document Vector Space Model

1. a hen lives on a farm.
2. a cow lives on a farm.

Term	Doc 1	Doc 2
a	2	2
lives	1	1
farm	1	1
hen	1	0
on	1	1
cow	0	1

**Table 4:** Term Count

- hen: [2, 1, 1, 1, 1, 0]
- cow: [2, 1, 1, 0, 1, 1]

**cosine similarity** - measure of similarity between two vectors

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

- Used to find similarity between documents
- $\text{cosineSimilarity}(\text{hen}, \text{cow}) = 0.87$ 
  - $\cos(0^\circ) = 1$  : *Similar*
  - $\cos(90^\circ) = 0$  : *Not similar*



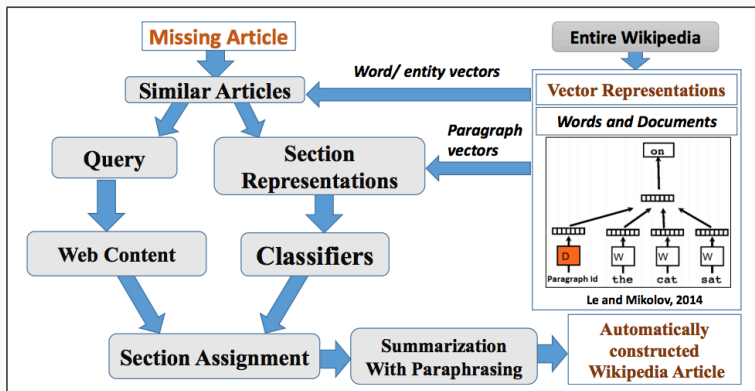
# Application

---

WikiWrite

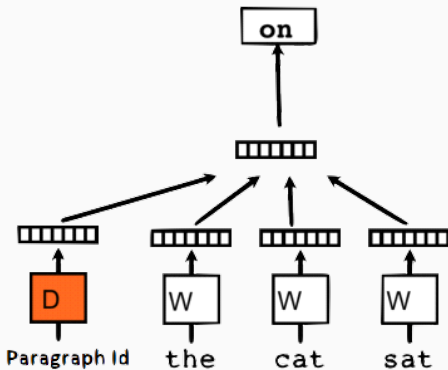
1. Current methods assume that the Wikipedia categories are known
2. Copyright violations
3. Coherence issues

# WikiWrite: Framework Overview



# WikiWrite: Paragraph Vector Distributed Memory (PVDM)

- Vector space model that preserves semantics
- Purpose for WikiWrite:
  1. Identification of similar articles on Wikipedia
  2. Inference of vector representations of new paragraphs retrieved from the web



## SpVgg Ostend 07 Frankfurt

From Wikipedia, the free encyclopedia

**Wikipedia does not have an article with this exact name.** Please [search for \*SpVgg Ostend 07 Frankfurt\* in Wikipedia](#) to check for alternative titles or spellings.

- [Log in or create an account](#) to start the ***SpVgg Ostend 07 Frankfurt*** article, alternatively use the [Article Wizard](#), or [add a request for it](#).
- [Search for "\*SpVgg Ostend 07 Frankfurt\*"](#) in existing articles.
- [Look for pages within Wikipedia that link to this title](#).

**Other reasons this message may be displayed:**

- If a page was recently created here, it may not be visible yet because of a delay in updating the database; wait a few minutes or try the [purge](#) function.
- Titles on Wikipedia are **case sensitive** except for the first character; please check [alternative capitalizations](#) and consider adding a [redirect](#) here to the correct title.
- If the page has been deleted, check the [deletion log](#), and see [Why was the page I created deleted?](#)


Look for **SpVgg Ostend 07 Frankfurt** on one of Wikipedia's [sister projects](#):

 [Wiktionary](#) (free dictionary)


 [Wikibooks](#) (free textbooks)

 [Wikiquote](#) (quotations)

 [Wikisource](#) (free library)

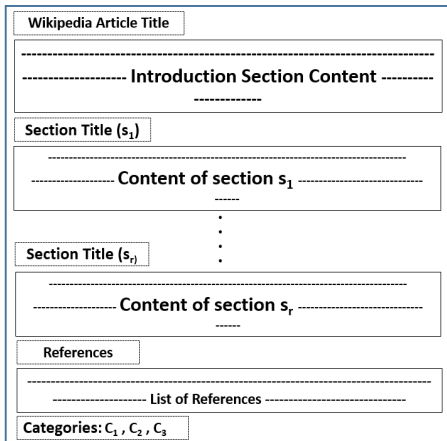
 [Wikiversity](#) (free learning resources)

 [Commons](#) (images and media)

 [Wikivoyage](#) (free travel guide)

 [Wikinews](#) (free news source)

 [Wikidata](#) (free linked database)



- Use PV-DM to find similar Wikipedia articles to the Red-linked entity
- Example - **Sonia Bianchetti**
  - Referee, International Skating Union (ISU), Judge, etc.

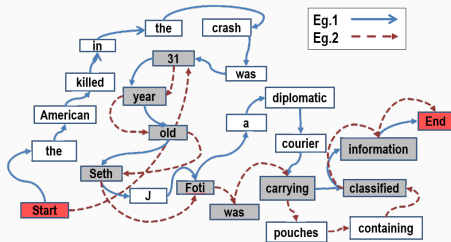
- **Query reformulation** - Rewrite query to be more specific.
- Example: Machine Learning → "Machine Learning" algorithm intelligence.
- Grab top 20 Google search results.

Wikipedia Article Title
----- ----- <b>Introduction Section Content</b> ----- -----
Section Title ( $s_1$ )
----- ----- <b>Google results 7, 16</b> ----- -----
⋮
Section Title ( $s_r$ )
----- ----- <b>Google results 2, 14</b> ----- -----
References
----- ----- <b>List of References</b> ----- -----
Categories: $C_1, C_2, C_3$

- Classifiers - Place query results in the right section of our new article.



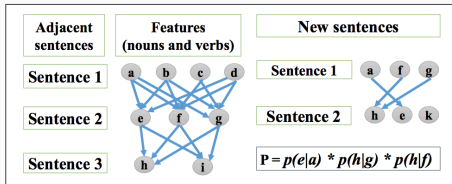
# WikiWrite: Summarization - Generate New Sentences



- A word-graph approach.
  - nodes: bigrams
  - edges: adjacency relationship between bigrams
- Cosine similarity  $\geq 0.8$  throw away
- Sentences with same parent sentence weighed heavily

- Sentence importance - cosine similarity between new sentence and reformulated query
- Linguistic quality - trigram language model find best sequence of words

# WikiWrite: Summarization - Coherence



- Adjacent sentence coherence to ensure global paragraph coherence
- **coherence score**
  - transition frequency  $\Rightarrow$  transition probability
- Multiply transition probabilities of individual features (nouns and verbs)
- Use cosine similarity to reduce Redundancy

- Paraphrase Database (PPDB)
- Make modifications to sentences and assign readability score

- “The NSSP initiative will lead to **significant economic** benefits for both countries”
  1. *significant economic* => *considerable economic*
  2. *economic benefits* => *financial advantages*
- Readability score for 1:
  - “lead to **considerable economic** benefit for”
- “The NSSP initiative will **result in major financial advantages** for the two countries ”

# Results

---

Summary Evaluation Metrics

# Summary Evaluation Metrics

- Recall-Oriented Understudy for Gisting Evaluation (ROUGE) - Measure n-gram overlap between generated summary and reference summaries.

$$ROUGE = \frac{\textit{n-gram match between system and references}}{\textit{n-grams in references}}$$

- F-Measure - Accuracy score

- ROUGE-N: N-gram based co-occurrence statistics



- ROUGE-N: N-gram based co-occurrence statistics
- ROUGE-L: LCS-based statistics

- ROUGE-N: N-gram based co-occurrence statistics
- ROUGE-L: LCS-based statistics
- ROUGE-S: Skip-bigram-based co-occurrence statistics
- ROUGE-W: Weighed version of ROUGE-L that favors consecutive LCSes

- Summary: police killed the gunman
  - $Ref_1$ : police kill the gunman
  - $Ref_2$ : the gunman kill police
1. ROUGE-2:  $Ref_1 = Ref_2$ 
    - “the gunman”

## ROUGE: Example ROUGE-L

- Summary: police killed the gunman
  - $Ref_1$ : police kill the gunman
  - $Ref_2$ : the gunman kill police
1. ROUGE-2:  $Ref_1 = Ref_2$ 
    - “the gunman”
  2. ROUGE-L:  $Ref_1 > Ref_2$ 
    - $Ref_1$ : “police the gunman”
    - $Ref_2$ : “the gunman”

# Results

---

## WikiWrite Results

- 1000 randomly selected popular articles
- Baseline systems:
  1. WikiKreator: assumes Wikipedia categories are known
  2. Perceptron-ILP: extractive
- Experiments:
  1. Section classification
  2. Content selection
  3. Generate new articles

- Predict the section title given the section content

**Table 5:** Section Classification Results

<b>Technique</b>	<b><math>F_1</math> Score</b>	<b>Average Time</b>
WikiWrite	0.622	~2 mins
WikiKreator	0.481	~10 mins

- Reconstruct Wikipedia articles using knowledge from the web
- WikiWrite (Ref) - doesn't use reformulated query

**Table 6:** Content Selection Results

Technique	ROUGE-1	ROUGE-2
WikiWrite	0.441	0.223
WikiWrite (Ref)	0.520	0.257
WikiKreator	0.371	0.183
Perceptron-ILP	0.342	0.169



- Generate new articles.

**Table 7:** 50 Generated Wikipedia articles

<b>Statistics</b>	
Number of articles in mainspace	47
Entire edit retained	12
Modification of content	35
Average number of edits	11
Percentage of references retained	72%

# Conclusion

- Abstractive summarization can be effective in generating Wikipedia articles
- Look into research ethics before committing: [link](#)
- Abstractive summarization attracting more researchers.
- Deep learning using neural networks is the future!

# Thanks!

Get the source of this theme and the demo presentation from

`github.com/matze/mtheme`

The theme *itself* is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



## Questions?

# References

1. Siddhartha Banerjee and Prasenjit Mitra. Wikiwrite: Generating wikipedia articles automatically. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9- 15 July 2016, pages 2740–2746, 2016.
2. Siddhartha Banerjee and Prasenjit Mitra. Wikikreator: Improving wikipedia stubs automatically. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pages 867–877, 2015.
3. Siddhartha Banerjee and Prasenjit Mitra. Filling the gaps: Improving wikipedia stubs. In Proceedings of the 2015 ACM Symposium on Document Engineering, DocEng '15, pages 117–120, New York, NY, USA, 2015. ACM.
4. Hongyan Jing. Using hidden markov modeling to decompose human-written summaries. *Comput. Linguist.*, 28(4):527–543, December 2002.
5. Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.