MASTER IN COMPUTER SCIENCE

Recommending ordered sections with similar section filtering to help structuring Wikipedia articles

Supervisor
Prof. Dr. Philippe Cudré-Mauroux
eXascale Infolab, Department of Informatics, University of Fribourg

Co-Supervisor
Natalia Ostapuk
eXascale Infolab, Department of Informatics, University of Fribourg

Sergiy Goloviatinski, University of Neuchâtel
Master thesis
03.02.2022

UNIVERSITÄT BERN

UNIVERSITÉ DE NEUCHÂTEL

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

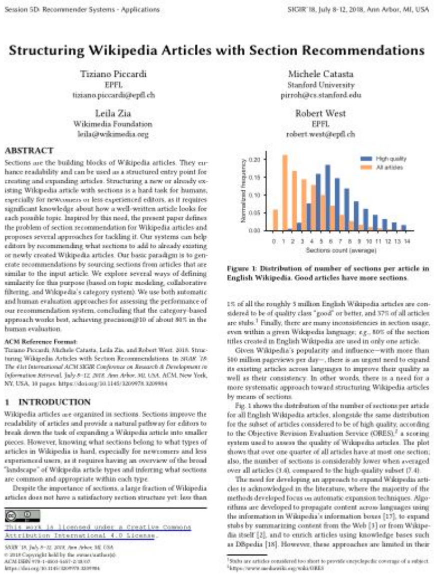
Outline

- Context and motivation
- Related work
- Improvements of existing method
- Redundant sections filtering
- Section ordering
- Prototype demo
- Conclusion

Context and motivation

- Wikipedia articles are structured with sections
- No official guidelines on structure, only few community made
  - → Need to look at examples
- Good quality articles have more sections (7.32 on average compared to 4.63 for all articles)
  - →More sections would improve quality
- 37% of articles are marked as stubs
- 87% of sections are used only in one article (e.g. “Early life of Dionysius the Elder”)
  - → Need for standardization
- →An algorithm for section recommendation would respond to an existing need

Structuring Wikipedia Articles with Section Recommendations



Structuring Wikipedia Articles with Section Recommendations

Tiziano Piccardi
EPFL
tiziano.piccardi@epfl.ch

Michele Catasta
Stanford University
pirroh@cs.stanford.edu

Leila Zia
Wikimedia Foundation
leila@wikimedia.org

Robert West
EPFL
robert.west@epfl.ch

ABSTRACT

Sections are the building blocks of Wikipedia articles. They enhance readability and can be used as a structured entry point for creating and expanding articles. Structuring a new or already existing Wikipedia article with sections is a hard task for humans, especially for newcomers or less experienced editors, as it requires significant knowledge about how a well-written article looks for each possible topic. Inspired by this need, the present paper defines the problem of section recommendation for Wikipedia articles and proposes several approaches for tackling it. Our systems can help editors by recommending what sections to add to already existing or newly created Wikipedia articles. Our basic paradigm is to generate recommendations by sourcing sections from articles that are similar to the input article. We explore several ways of defining similarity for this purpose (based on topic modeling, collaborative filtering, and Wikipedia's category system). We use both automatic and human evaluation approaches for assessing the performance of our recommendation system, concluding that the category-based approach works best, achieving precision@10 of about 80% in the human evaluation.

ACM Reference Format:
Tiziano Piccardi, Michele Catasta, Leila Zia, and Robert West. 2018. Structuring Wikipedia Articles with Section Recommendations. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3209984

1 INTRODUCTION

Wikipedia articles are organized in sections. Sections improve the readability of articles and provide a natural pathway for editors to break down the task of expanding a Wikipedia article into smaller pieces. However, knowing what sections belong to what types of articles in Wikipedia is hard, especially for newcomers and less experienced users, as it requires having an overview of the broad "landscape" of Wikipedia article types and inferring what sections are common and appropriate within each type.

Despite the importance of sections, a large fraction of Wikipedia articles does not have a satisfactory section structure yet: less than

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07.
https://doi.org/10.1145/3209978.3209984

Figure 1: Distribution of number of sections per article in English Wikipedia. Good articles have more sections.

1% of all the roughly 5 million English Wikipedia articles are considered to be of quality class "good" or better, and 37% of all articles are stubs.[1] Finally, there are many inconsistencies in section usage, even within a given Wikipedia language; e.g., 80% of the section titles created in English Wikipedia are used in only one article.

Given Wikipedia's popularity and influence—with more than 500 million pageviews per day—, there is an urgent need to expand its existing articles across languages to improve their quality as well as their consistency. In other words, there is a need for a more systematic approach toward structuring Wikipedia articles by means of sections.

Fig. 1 shows the distribution of the number of sections per article for all English Wikipedia articles, alongside the same distribution for the subset of articles considered to be of high quality, according to the Objective Revision Evaluation Service (ORES),[2] a scoring system used to assess the quality of Wikipedia articles. The plot shows that over one quarter of all articles have at most one section; also, the number of sections is considerably lower when averaged over all articles (3.4), compared to the high-quality subset (7.4).

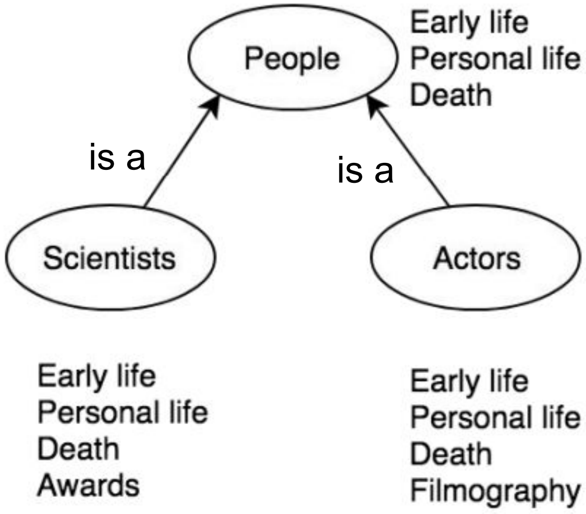
The need for developing an approach to expand Wikipedia articles is acknowledged in the literature, where the majority of the methods developed focus on automatic expansion techniques. Algorithms are developed to propagate content across languages using the information in Wikipedia's information boxes [17], to expand stubs by summarizing content from the Web [3] or from Wikipedia itself [2], and to enrich articles using knowledge bases such as DBpedia [18]. However, these approaches are limited in their

[1] Stubs are articles considered too short to provide encyclopedic coverage of a subject.
[2] https://www.mediawiki.org/wiki/ORES

- Use Wikipedia category network
- Recommended sections → ranked by P(Section | Category)

Tiziano Piccardi, et al. Structuring wikipedia articles with section recommendations. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval,* 2018 https://dl.acm.org/doi/pdf/10.1145/3209978.3209984

Structuring Wikipedia Articles with Section Recommendations: Idea



- Use Wikipedia category network as ontology
- A scientist is a person
- An actor is also a person
- Therefore, sections from articles about scientists and actors can be recommended for articles about persons

Image source:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Structuring Wikipedia Articles with Section Recommendations: Problem

Category:French artists

From Wikipedia, the free encyclopedia

Subcategories

- French cartoonists (4 C, 110 P)
- French ceramists (2 C, 31 P)
- French cinematographers (3 C, 117 P)
- French comics artists (3 C, 168 P)
- French conceptual artists (28 P)
- French contemporary artists (1 C, 164 P)

Categories: French people in arts occupations | Artists by nationality | French art

- Wikipedia category network is not guaranteed to be an ontology

is a(french cartoonist, french artist)

is a (french artists, french people in arts occupations)

is a(french artists, french art)

Structuring Wikipedia Articles with Section Recommendations: Method (1)

55 DBpedia types


About: Pablo Picasso

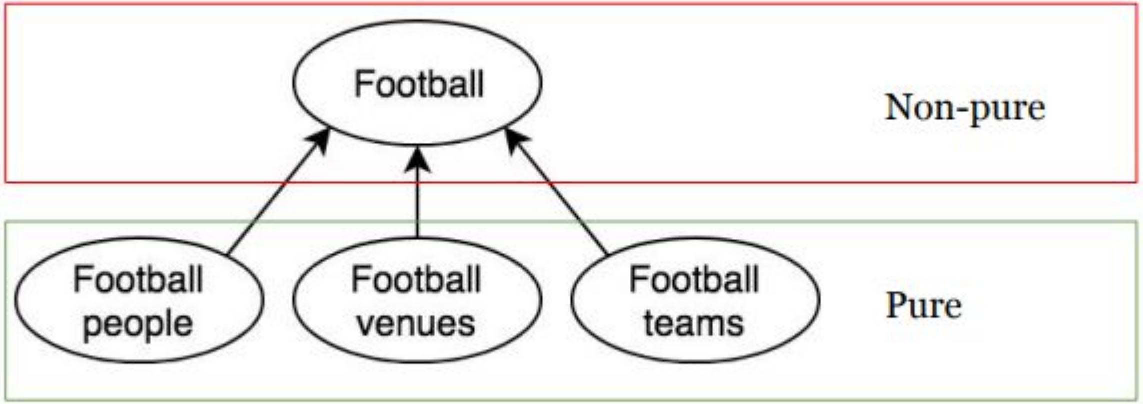
An Entity of Type: person, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

Pablo Ruiz Picasso (25 October 1881 – 8 April 1973) was a Spanish painter, sculptor, printmaker, ceramicist and theatre designer who spent most of his adult life in France. Regarded as one of the most influential artists of the 20th century, he is known for co-founding the Cubist movement, the invention of constructed sculpture, the co-invention of collage, and for the wide variety of styles that he helped develop and explore. Among his most famous works are the proto-Cubist Les Demoiselles d'Avignon (1907), and Guernica (1937), a dramatic portrayal of the bombing of Guernica by German and Italian air forces during the Spanish Civil War.

- Need a way to measure if a category relationship is ontological
- Using an external source: DBpedia
- Entities on DBpedia about Wikipedia articles
- Use type attribute

Structuring Wikipedia Articles with Section Recommendations: Method (2)

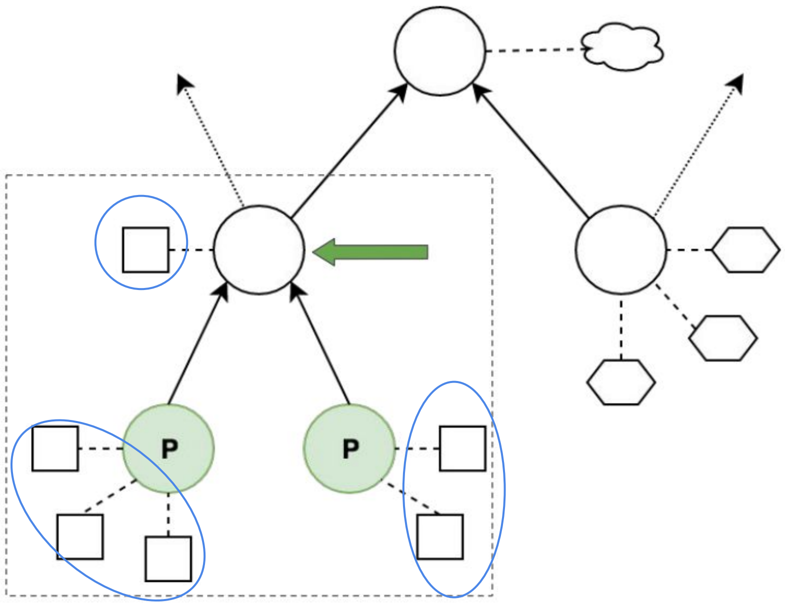
Person + Location + Organization

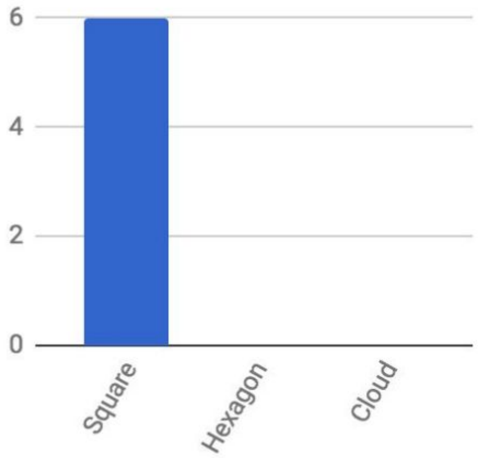


Person Location Organization

Image source:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Structuring Wikipedia Articles with Section Recommendations: Method (3a)



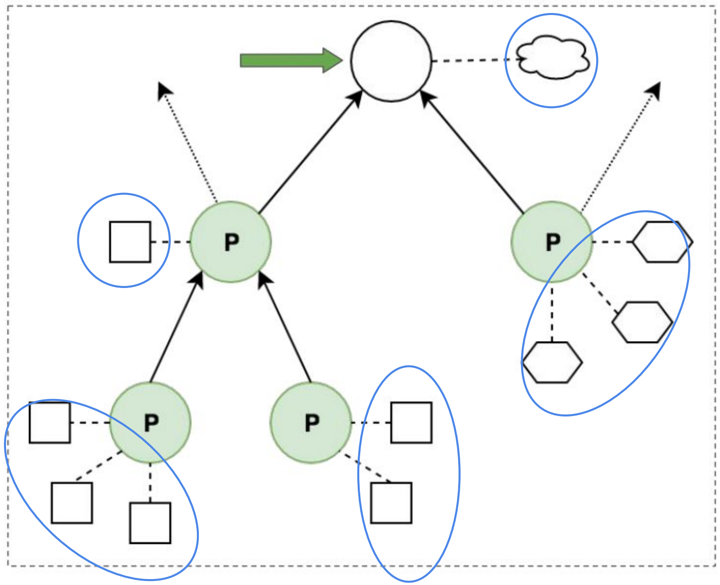


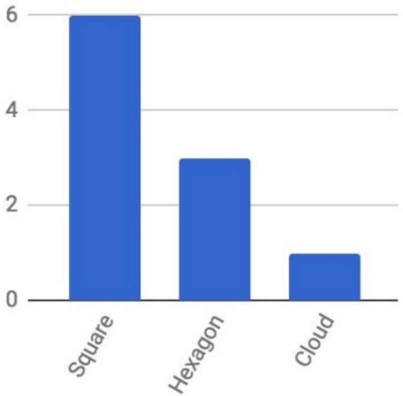
Article type distribution

- Mark which categories are pure, based on article type distribution
- Bottom-up approach
- If pure, add articles to parent
- Continue while categories pure in the hierarchy

Image source:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Structuring Wikipedia Articles with Section Recommendations: Method (3b)



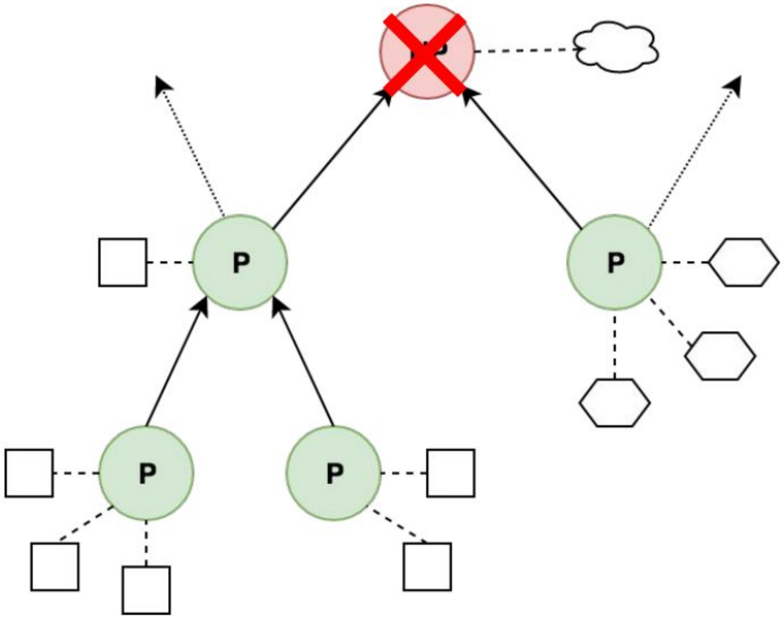


Article type distribution

- Because child categories are pure, add their articles to parent category
- Article type distribution of parent contains articles from children
- But now, there are too many different types in the parent category

Image source:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Structuring Wikipedia Articles with Section Recommendations: Method (3c)

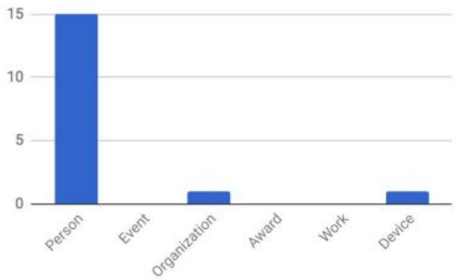


- The category is marked as not pure, we prune the graph

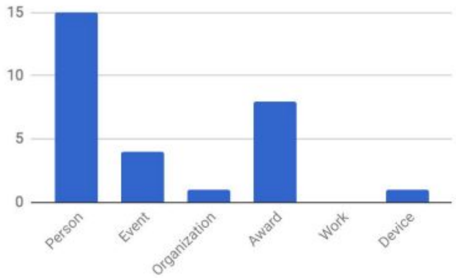
Image source:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Structuring Wikipedia Articles with Section Recommendations: Method (4)

Gini coefficient to quantify purity of article type distribution: minimum 0.966



gini ≥ 0.966



gini < 0.966

Image sources:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Backup slide: gini coefficient

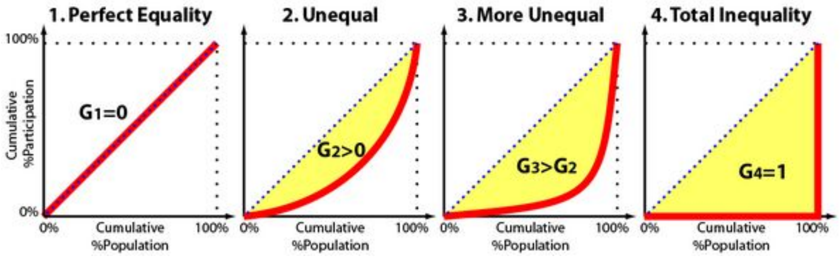


Image sources:
https://towardsdatascience.com/clearly-explained-gini-coefficient-and-lorenz-curve-fe6f5dcdc07

Structuring Wikipedia Articles with Section Recommendations: Pipeline

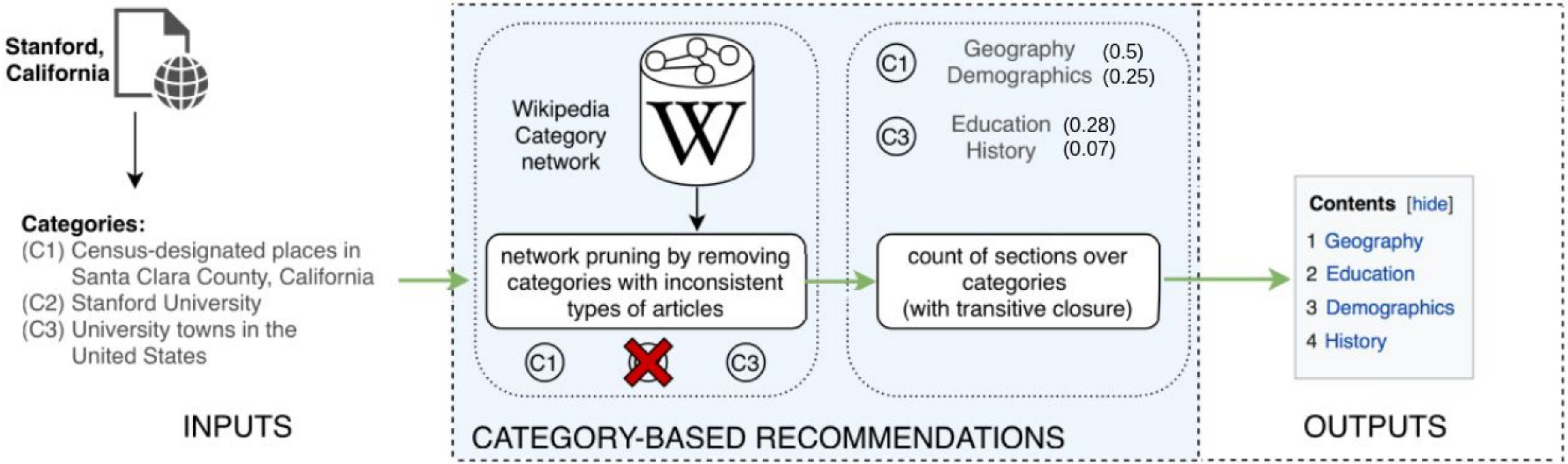


Image source:
Tiziano Piccardi, et al. Structuring wikipedia articles with section recommendations.
.*The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018
https://dl.acm.org/doi/pdf/10.1145/3209978.3209984

Structuring Wikipedia Articles with Section Recommendations: Limitations

Semantically redundant sections

1. Gameplay
2. Story
3. Plot
4. Development and release
5. Development
6. Reception
7. Sequel

Article about video game

No logical section ordering

1. First round (0.8)
2. Second round (0.8)
3. Third round (0.8)
4. Final (0.8)
5. Fourth round (0.6)
6. Quarter-finals (0.6)
7. Semi-finals (0.6)
8. Fifth round (0.4)

P(Section | Category)

Article about Scottish football cup

Improvement of existing method

Improvement of existing method: unknown article types (1)

?

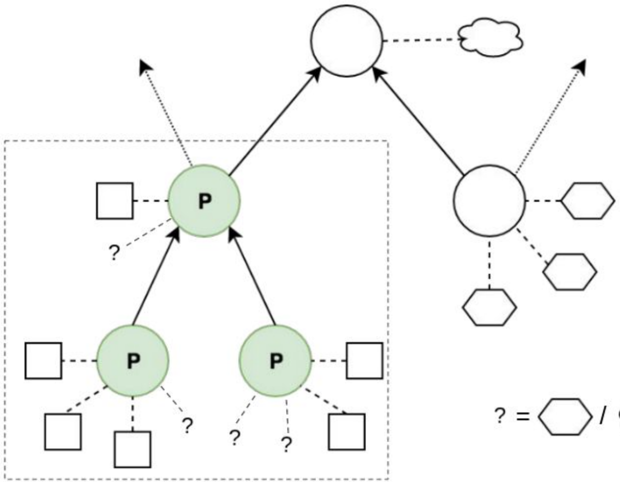
About: Albedo

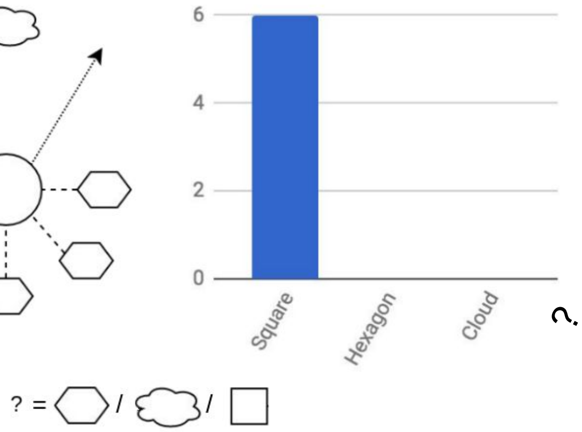
An Entity of Type: Thing, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

Albedo (/ælˈbiːdoʊ/; from Latin albedo 'whiteness') is the measure of the diffuse reflection of solar radiation out of the total solar radiation and measured on a scale from 0, corresponding to a black body that absorbs all incident radiation, to 1, corresponding to a body that reflects all incident radiation. The term albedo was introduced into optics by Johann Heinrich Lambert in his 1760 work Photometria.

- “Thing” is too generic for a type, not used in method
- Therefore we don’t know the type of the article

Improvement of existing method: unknown article types (2)





- Unknown article types not included in the article type distribution
- Those articles could be of any type
- Categories containing unknown article types are marked as pure, but maybe it would not be the case if those article types would be known

Image source:
https://upload.wikimedia.org/wikipedia/commons/c/cb/Using_Wikipedia_categories_for_research.pdf

Improvement of existing method: example of noise in recommendations

1. Early life
2. Biography
3. Career
4. History
5. Death

Article about "Monument types"

- Sections in red not relevant for monuments
- Those sections in red come from the category "Names inscribed under the Arc de Triomphe"

Solution

- Filter out unknown types articles from dataset
- Reduced number of articles from ~3.6 million to ~2 million
- 56% of articles remained

1. Early life
2. Biography
3. Career
4. History
5. Death

remove articles with unknown types →

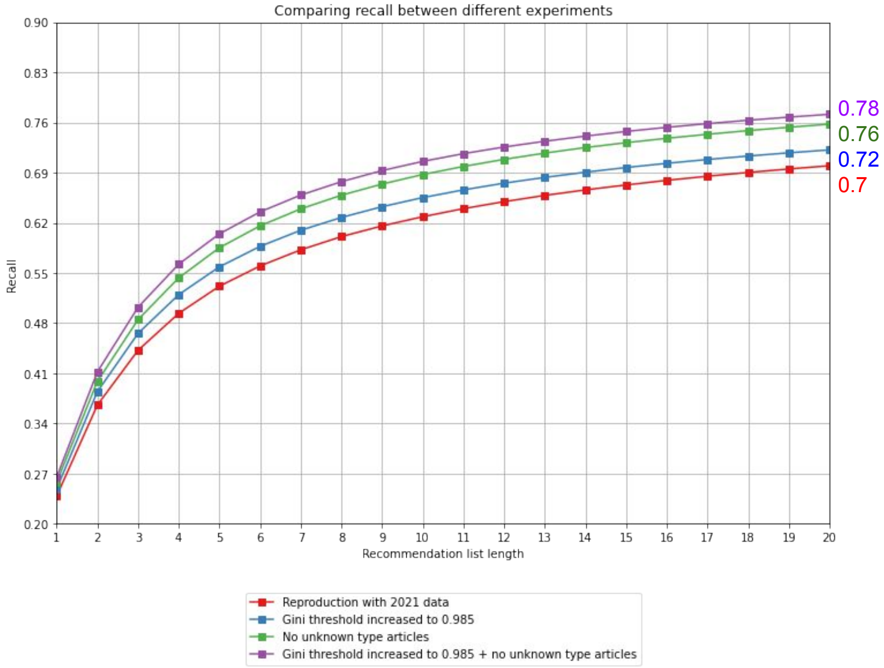
1. History
2. Gallery
3. Architecture
4. Location
5. Description

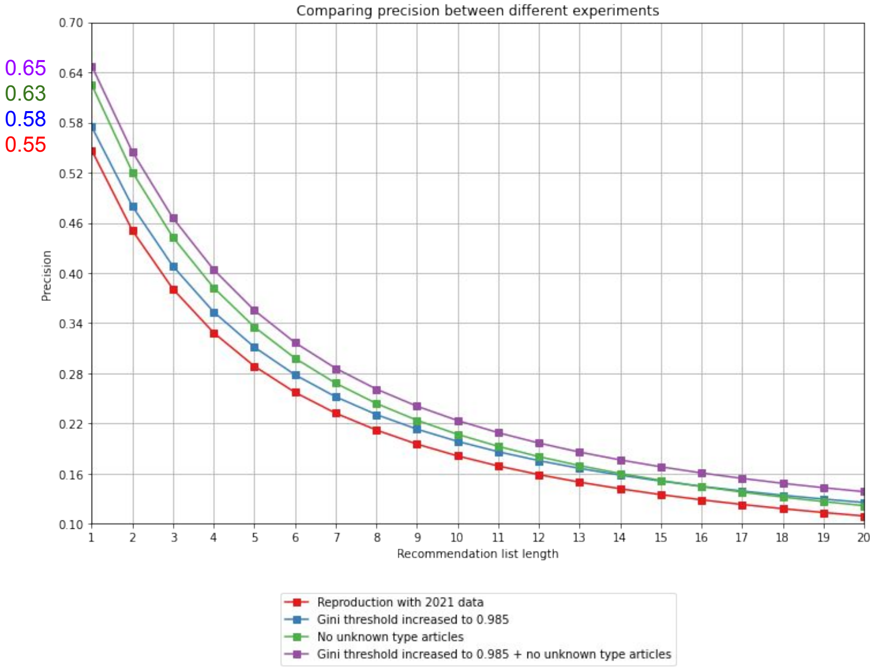
Article about “Monument types”

(backup slide) Improvement to existing method: increasing gini threshold

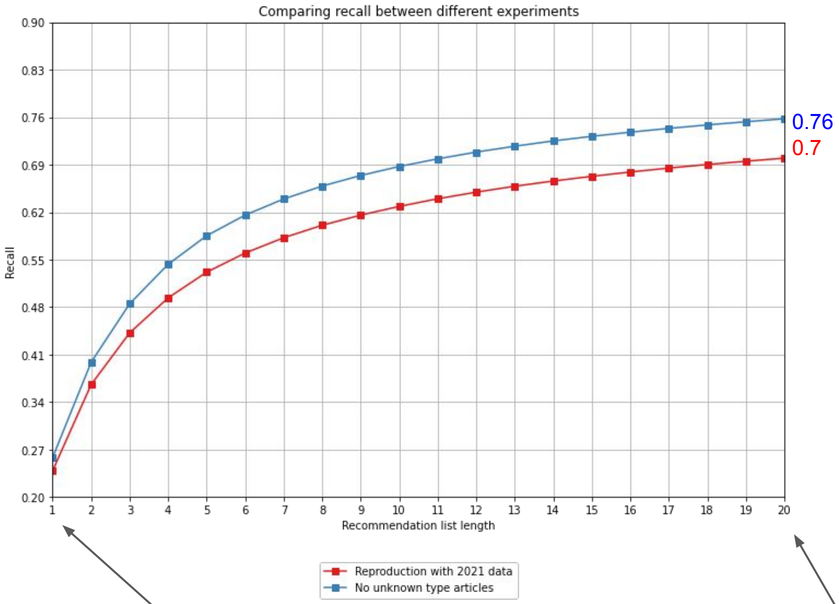
- Gini threshold increased from 0.966 to 0.985 gave us a better performance

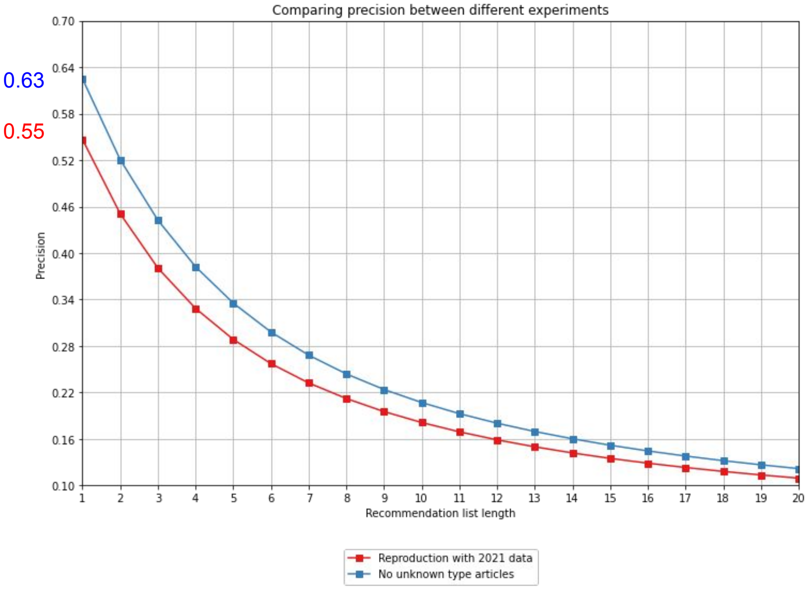
(backup slide) Existing method improvement: results





Existing method improvement: results





most frequent section in category

20 most frequent sections in category

Redundant sections filtering

Redundant sections filtering: idea

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

Nils Reimers and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) has set a new state-of-the-art performance on sentence-pair regression tasks like semantic textual similarity (STS). However, it requires that both sentences are fed into the network, which causes a massive computational overhead: Finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations (~65 hours) with BERT. The construction of BERT makes it unsuitable for semantic similarity search as well as for unsupervised tasks like clustering.

In this publication, we present Sentence-BERT (SBERT), a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT.

We evaluate SBERT and SRoBERTa on common STS tasks and transfer learning tasks, where it outperforms other state-of-the-art sentence embeddings methods.[1]

1 Introduction

In this publication, we present Sentence-BERT (SBERT), a modification of the BERT network using siamese and triplet networks that is able to derive semantically meaningful sentence embeddings[2]. This enables BERT to be used for certain new tasks, which up-to-now were not applicable for BERT. These tasks include large-scale semantic similarity comparison, clustering, and information retrieval via semantic search.

BERT set new state-of-the-art performance on various sentence classification and sentence-pair regression tasks. BERT uses a cross-encoder: Two sentences are passed to the transformer network and the target value is predicted. However, this setup is unsuitable for various pair regression tasks due to too many possible combinations. Finding in a collection of $n = 10\,000$ sentences the pair with the highest similarity requires with BERT $n \cdot (n-1)/2 = 49\,995\,000$ inference computations. On a modern V100 GPU, this requires about 65 hours. Similar, finding which of the over 40 million existent questions of Quora is the most similar for a new question could be modeled as a pair-wise comparison with BERT, however, answering a single query would require over 50 hours.

A common method to address clustering and semantic search is to map each sentence to a vector space such that semantically similar sentences are close. Researchers have started to input individual sentences into BERT and to derive fixed-size sentence embeddings. The most commonly used approach is to average the BERT output layer (known as BERT embeddings) or by using the output of the first token (the [CLS] token). As we will show, this common practice yields rather bad sentence embeddings, often worse than averaging GloVe embeddings (Pennington et al., 2014).

To alleviate this issue, we developed SBERT. The siamese network architecture enables that fixed-sized vectors for input sentences can be derived. Using a similarity measure like cosine-similarity or Manhatten / Euclidean distance, semantically similar sentences can be found. These similarity measures can be performed extremely efficient on modern hardware, allowing SBERT to be used for semantic similarity search as well as for clustering. The complexity for finding the

[1] Code available: https://github.com/UKPLab/sentence-transformers

[2] With semantically meaningful we mean that semantically similar sentences are close in vector space.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*
https://arxiv.org/abs/1908.10084

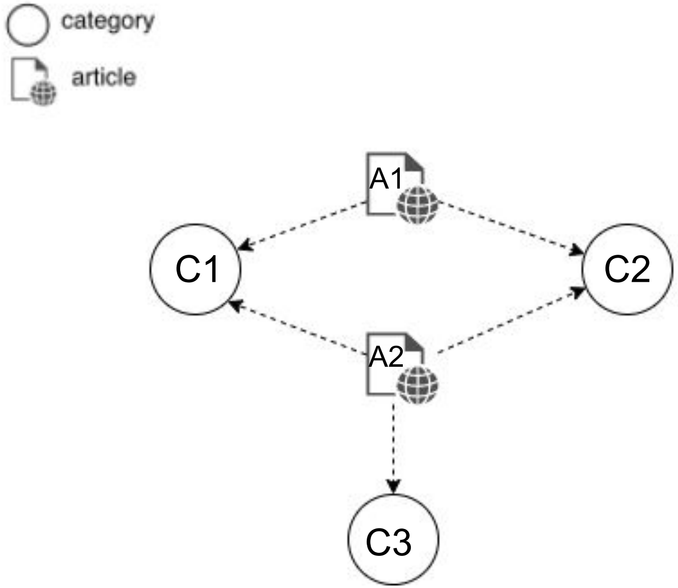
Redundant sections = semantically similar section contents

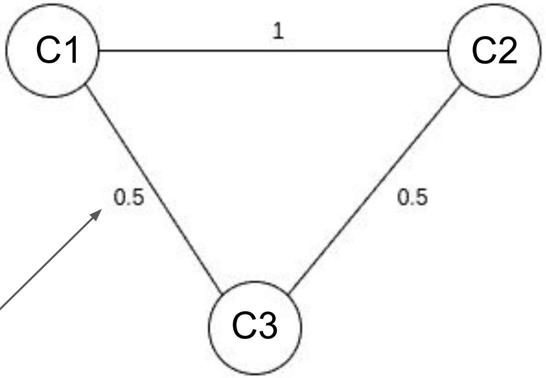
semantic similarity(Plot, Synopsis) = cosine similarity(A, B)

Redundant sections filtering: problem

- Which sections from which article to compare ?
- e.g. “History” section has different content depending on context
- Group categories which are likely to have articles in common
- →group sections that are likely to appear together in recommendation lists
- Find semantically similar sections from articles in each group

(backup slide) Redundant sections filtering: group categories by context (1)

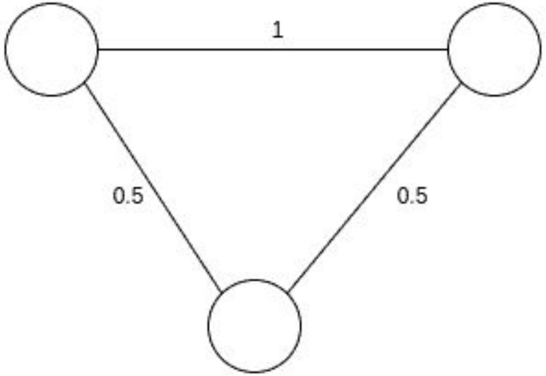




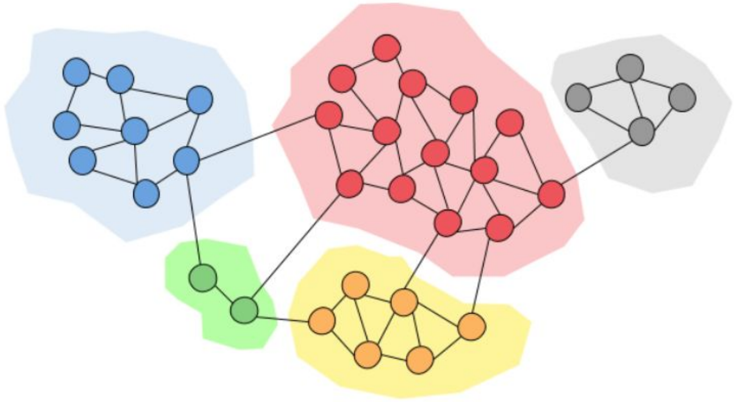
$$\frac{|Articles(C1) \cap Articles(C3)|}{|Articles(C1) \cup Articles(C3)|} = \frac{|\{A2\}|}{|\{A1, A2\}|} = \frac{1}{2}$$

Articles(x): articles belonging to category x

(backup slide) Redundant sections filtering: group categories by context (2)

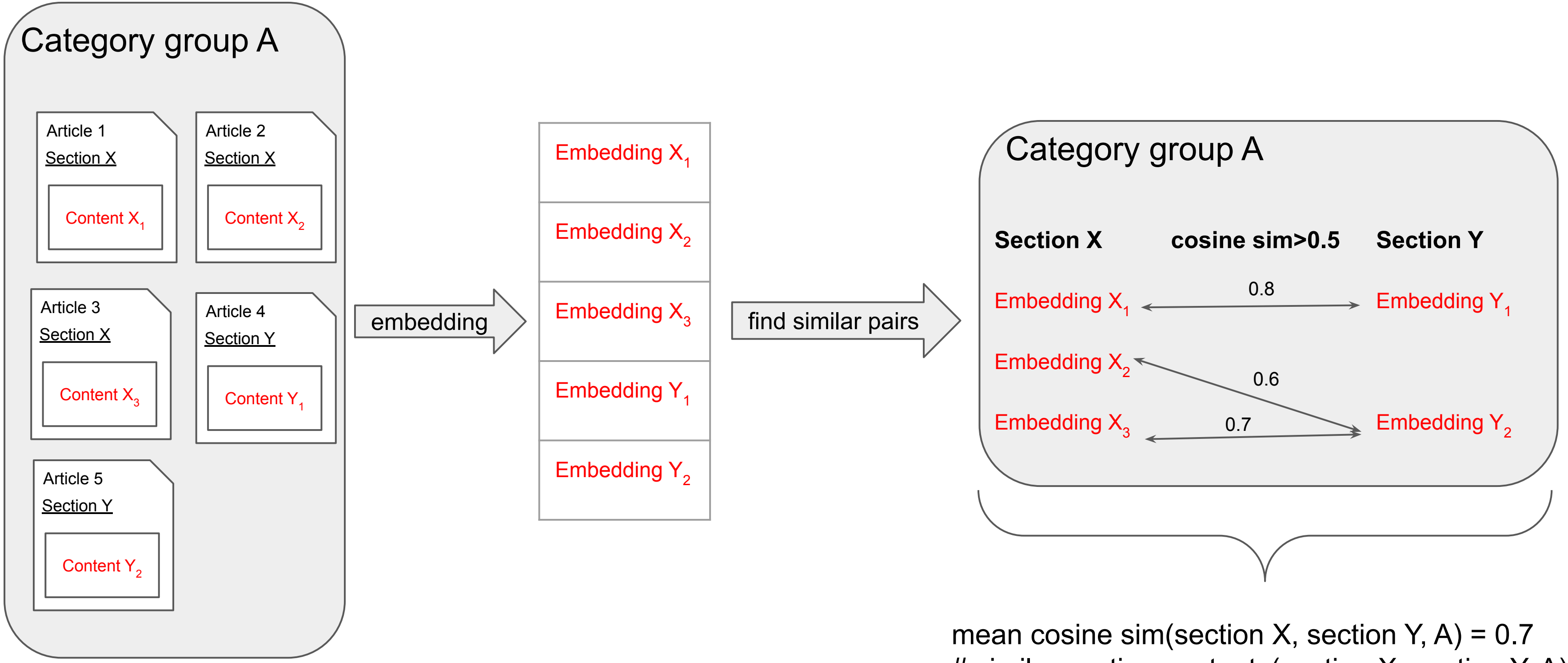


Community detection



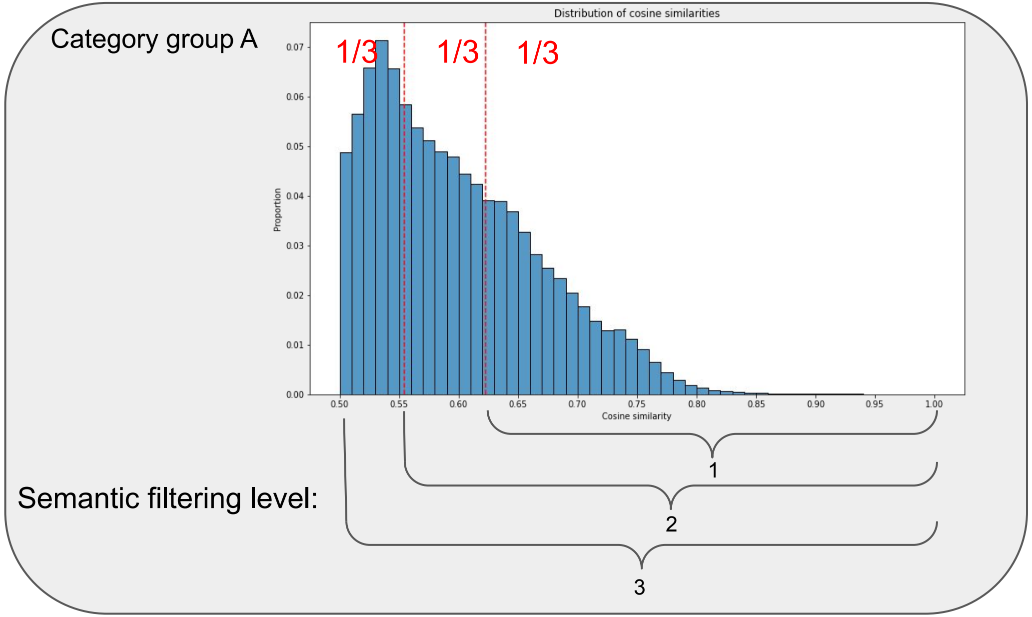
Louvain method

Redundant sections filtering: method (1)



mean cosine sim(section X, section Y, A) = 0.7
similar section contents(section X, section Y, A) = 3

Redundant sections filtering: method (2)



- Distribution of cosine similarities different in each category group
- Defined semantic filtering level to decide if redundant sections will be filtered
- Given level corresponds to a cosine similarity threshold different in each category group
- Section pairs with cosine similarity above threshold considered as redundant

- Level 0: no filtering
- Level 1: filter top 1/3 most similar
- Level 2: filter top 2/3 most similar
- Level 3: filter all sections detected as similar

Redundant sections filtering: method (3)

Similar section pairs from same category group with cosine sim > threshold given by semantic filtering level

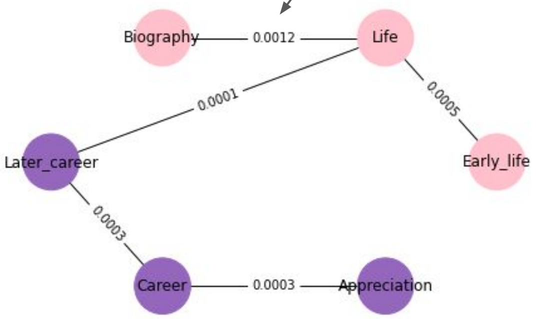
1. Early life
2. Biography
3. Life
4. Career
5. Personal
6. Later career
7. Appreciation

Article about classical musician

Pairs of semantically similar sections

(Biography, Life)
(Life, Early Life)
(Later career, Life)
(Career, Later career)
(Career, Appreciation)

$$\frac{\#\,similar\ section\ contents(Biography, Life)}{\#\,section\ contents(Biography)\cdot\#\,section\ contents(Life)}$$



Community detection with Louvain method

1. Biography
2. Career
3. Personal

Keep section x in each similar section group with highest
$$\#\,section\ contents(x)\cdot mean(edge\ weights(x))$$

Redundant sections filtering: example

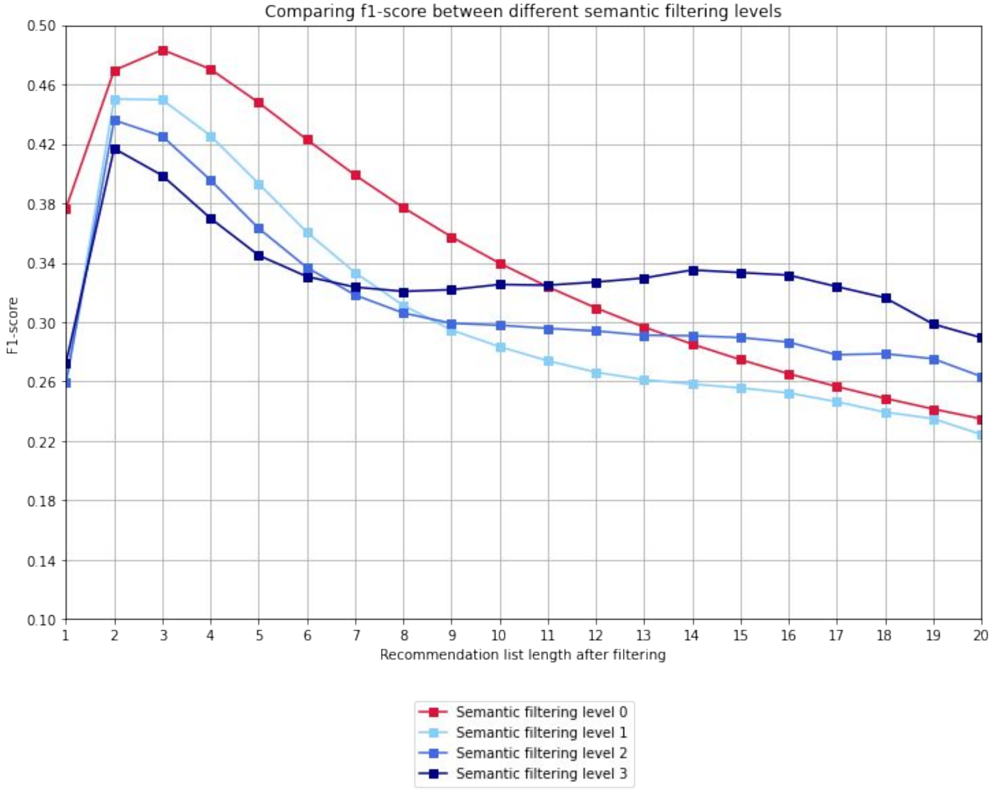
1. Gameplay
2. Story
3. Plot
4. Development and release
5. Development
6. Reception
7. Sequel

Article about video game

redundant sections filtering

1. Gameplay
2. Plot
3. Development
4. Reception
5. Sequel

Redundant sections filtering: results

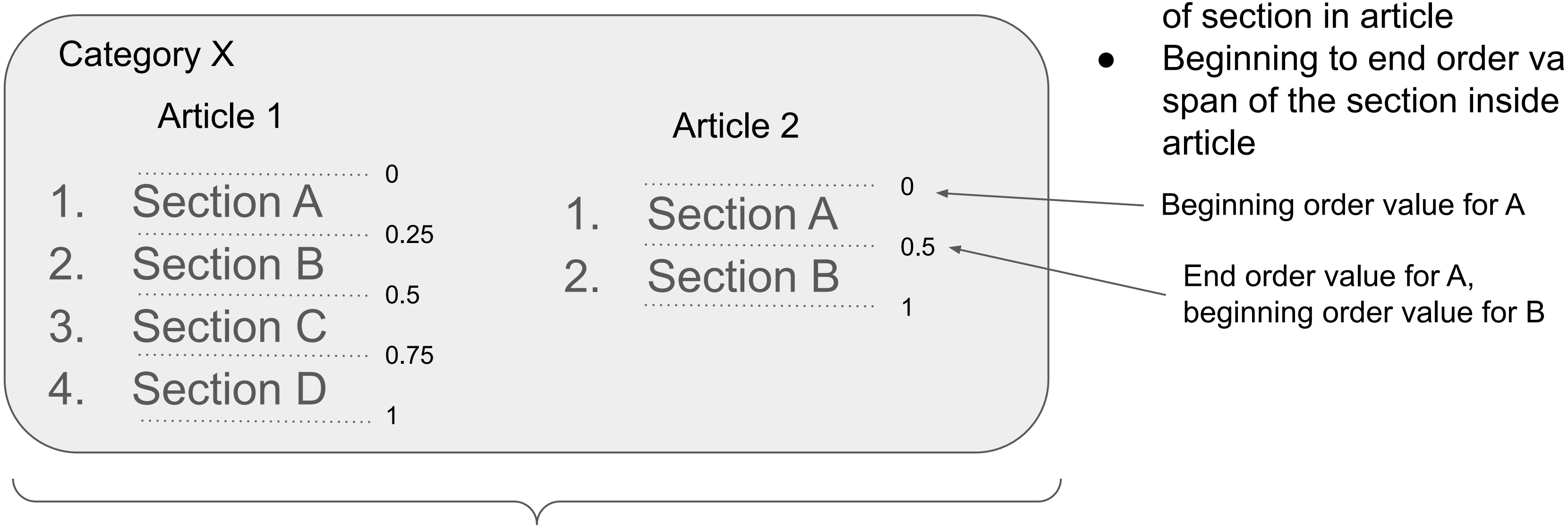


0.29
0.26
0.24
0.22

Section ordering

Section ordering: method (1)

- Order value: relative position of section in article
- Beginning to end order value: span of the section inside article



Average beginning and end order values by section in each category:
Beginning order value(A) = (0 + 0)/2 = 0
End order value(A) = (0.25 + 0.5)/2 = 0.375

Section ordering: method (2)

begin(x) : average beginning order value of section x
end(x) : average end order value of section x

Category 1

begin(A) = 0
end(A) = 0.2

begin(B) = 0.1
end(B) = 0.3

Category 2

begin(B) = 0.2
end(B) = 0.6

begin(C) = 0.5
end(C) = 1

$$midpoint(x) = \frac{begin(x) + end(x)}{2}$$

Recommended sections

- Section B
- Section C
- Section A

X	begin(X)	end(X)	midpoint(X)
B	(0.1+0.2)/2 = 0.15	(0.3+0.6)/2 = 0.45	(0.15+0.45)/2 = 0.3
C	0.5	1	(0.5+1)/2 = 0.75
A	0	0.2	(0+0.2)/2 = 0.1

Ordered recommended sections

1. Section A (0.1)
2. Section B (0.3)
3. Section C (0.75)

Section ordering: example

ordered by P(Section | Category)

1. First round (0.8)
2. Second round (0.8)
3. Third round (0.8)
4. Final (0.8)
5. Fourth round (0.6)
6. Quarter-finals (0.6)
7. Semi-finals (0.6)
8. Fifth round (0.4)

ordering

ordered by midpoint(Section)

1. First round (0.22)
2. Second round (0.33)
3. Third round (0.44)
4. Fourth round (0.53)
5. Fifth round (0.59)
6. Quarter-finals (0.7)
7. Semi-finals (0.81)
8. Final (0.9)

Article about Scottish football cup

(backup slide) Section ordering: evaluation, kendall's tau

Automatic Evaluation of Information Ordering: Kendall's Tau

Mirella Lapata*
University of Edinburgh

This article considers the automatic evaluation of information ordering, a task underlying many text-based applications such as concept-to-text generation and multidocument summarization. We propose an evaluation method based on Kendall's τ, a metric of rank correlation. The method is inexpensive, robust, and representation independent. We show that Kendall's τ correlates reliably with human ratings and reading times.

1. Introduction

The systematic evaluation of natural language processing (NLP) systems is an important prerequisite for assessing their quality and improving their performance. Traditionally, human involvement is called for in evaluating systems that generate textual output. Examples include text generation, summarization, and, notably, machine translation. Human evaluations consider many aspects of automatically generated texts ranging from grammaticality to content selection, fluency, and readability (Teufel and van Halteren 2004; Nenkova 2005; Mani 2001; White and O'Connell 1994).

The relatively high cost of producing human judgments, especially when evaluations must be performed quickly and frequently, has encouraged many researchers to seek ways of evaluating system output automatically. Papineni et al. (2002) proposed BLEU, a method for evaluating candidate translations by comparing them against reference translations (using n-gram co-occurrence overlap). Along the same lines, the content of a system summary can be assessed by measuring its similarity to one or more manual summaries (Hovy and Lin 2003). Bangalore, Rambow, and Whittaker (2000) introduce a variety of quantitative measures for evaluating the accuracy of an automatically generated sentence against a reference corpus string.

Despite differences in application and form, automatic evaluation methods usually involve the following desiderata. First, they measure numeric similarity or closeness of system output to one or several gold standards. Second, they are inexpensive, robust, and ideally language independent. Third, correlation with human judgments is an important part of creating and testing an automated metric. For instance, several studies have shown that BLEU correlates with human ratings on machine translation quality (Papineni et al. 2002; Doddington 2002; Coughlin 2003). Bangalore, Rambow, and Whittaker (2000) demonstrate that tree-based evaluation metrics for

Mirella Lapata. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics, Volume 32, Number 4*, 2006
https://aclanthology.org/J06-4002/

$$\tau = 1 - \frac{2 \cdot I}{0.5 \cdot n(n-1)}$$

with:

- I : nb of intersections
- n : nb of elements

A = [1,2,3,4]

B = [1,3,2,4]

C = [4,3,2,1]

tau(A,A) = 1

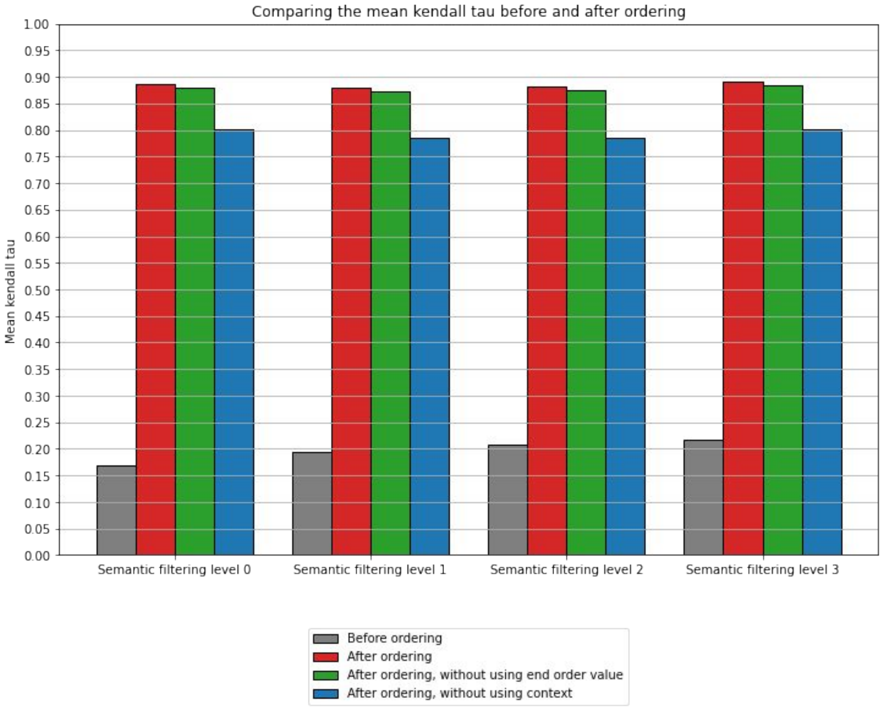
tau(A,B) = $1 - \frac{2 \cdot 1}{0.5 \cdot 4(4-1)} = 1 - \frac{2}{6} = \frac{2}{3}$

tau(A,C) = -1

(backup slide) Section ordering: experiments

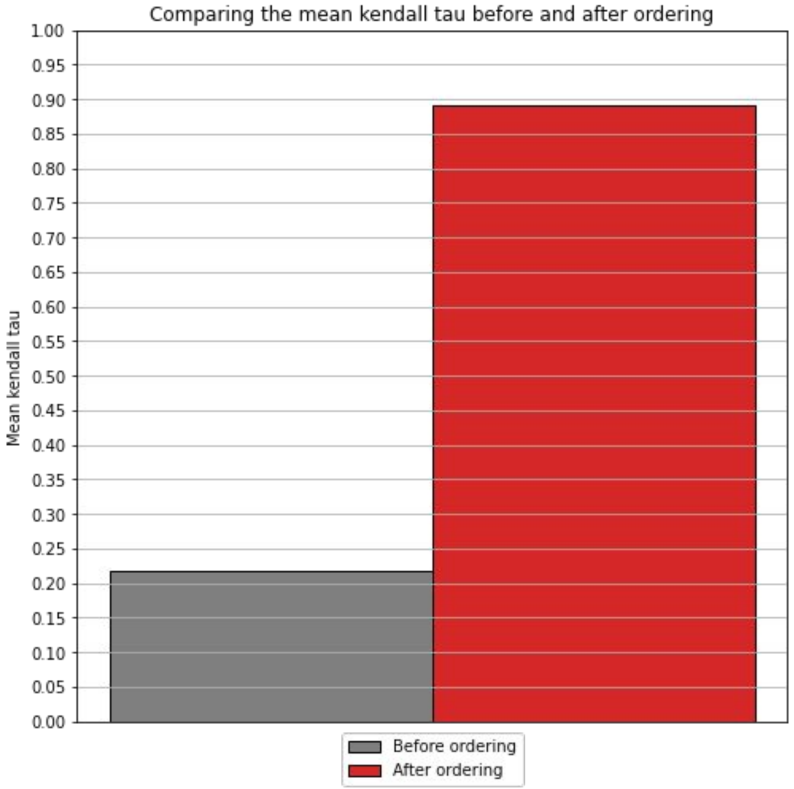
- Order by ascending mean(beginning order value,end order value), different for each category
- Order by ascending beginning order value, different for each category
- Order by ascending mean(beginning order value,end order value), among all categories (as if all articles in same category)

(backup slide) Section ordering: results for different experiments



Section ordering: results

Averaged for recommendation lists of size 20 which had at least 2 sections in common with ground truth



Prototype demo

http://127.0.0.1:5000

Conclusion

- Improved performance of existing method (precision@1 0.55→0.65 recall@20 0.7 →0.77)
- Added two features
  - Redundant section filtering (f1 score@20 0.24→0.29)
  - Order sections logically (average Kendall’s tau 0.22→0.89)
- Implemented prototype to demonstrate use case