DocEng 2015
15th ACM SIGWEB International Symposium on Document Engineering
Session 3: Layouts Improved

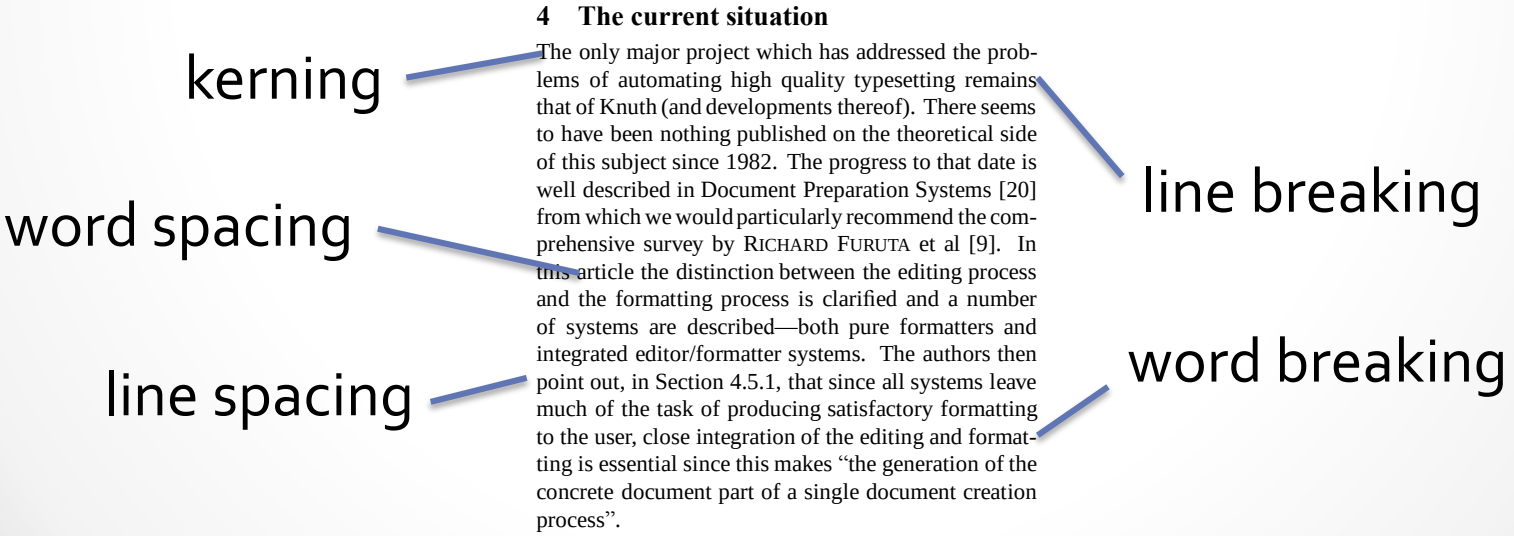
Document Layout Optimization with Automated Paraphrasing

Yusuke Kido
The University of Tokyo, Japan

Hikaru Yokono　Goran Topić　Akiko Aizawa
National Institute of Informatics, Japan

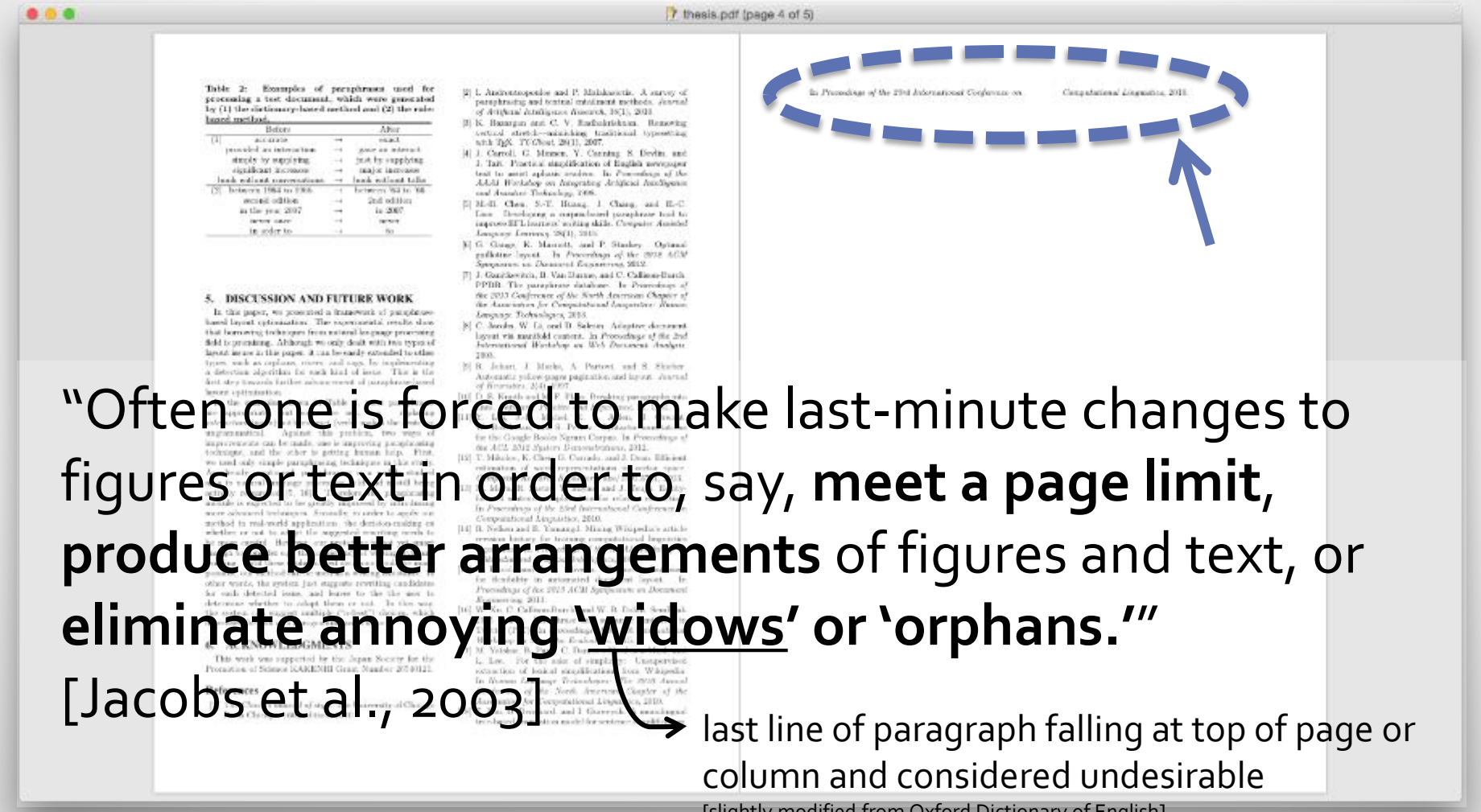
Document Layout Optimization

Wide-ranging problems related to **editing documents**, **typesetting**, or **rendering documents**



Mittelbach, F., & Rowley, C. (2004). The pursuit of quality. Health Affairs, 23(2), 102.

Limitation



"Often one is forced to make last-minute changes to figures or text in order to, say, **meet a page limit**, **produce better arrangements** of figures and text, or **eliminate annoying 'widows' or 'orphans.'"**
[Jacobs et al., 2003]

last line of paragraph falling at top of page or column and considered undesirable
[slightly modified from Oxford Dictionary of English]

Limitation

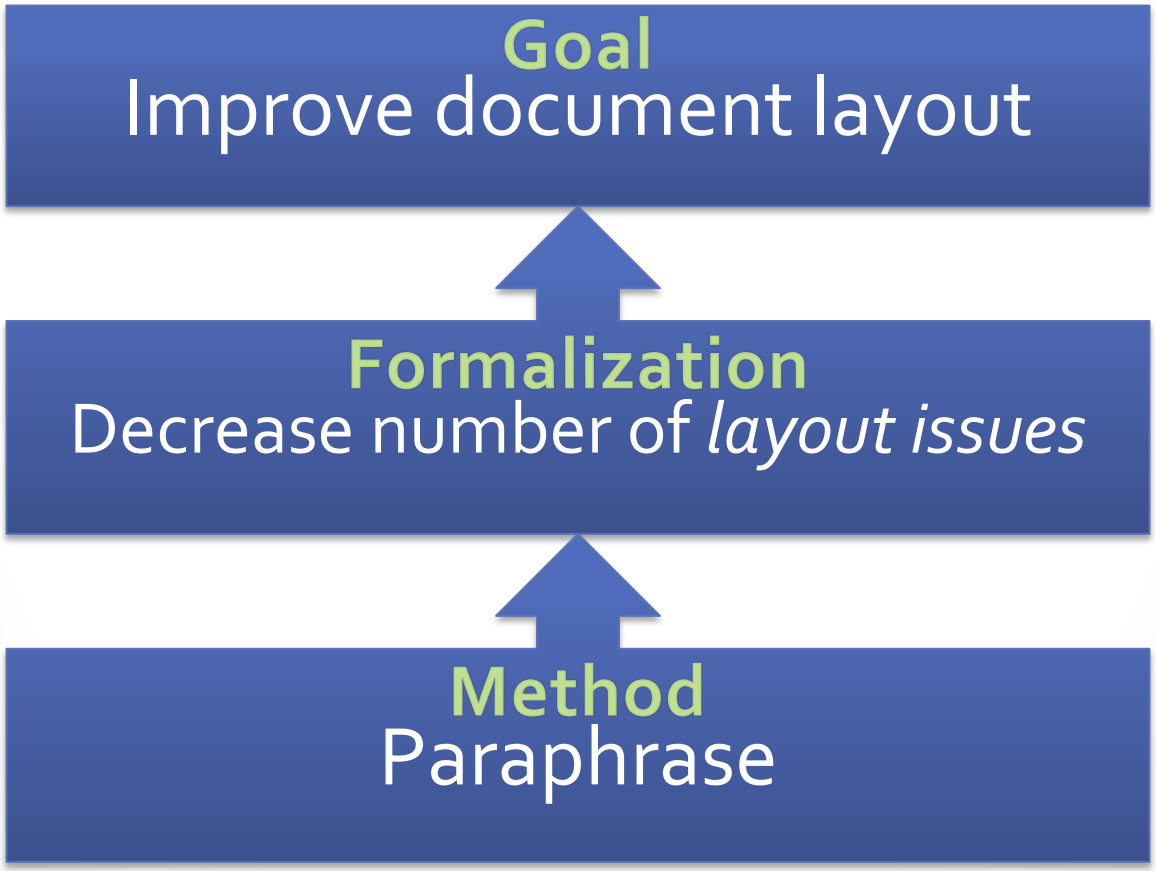
Computer is better at solving document layout optimization problems (e.g. line-breaking problem) than human "unless we give this person the liberty to **change the wording in order to obtain a better fit**"

[Knuth and Plass, 1981]

Human can change wording; how about computer?

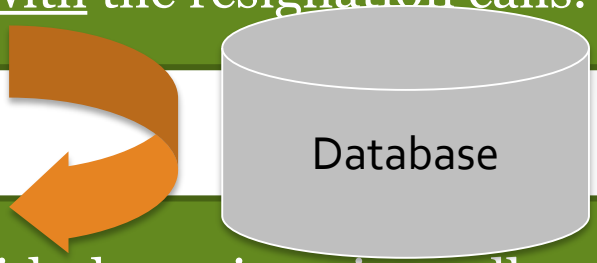
➜ **Natural Language Processing** (NLP)

Idea


Goal
Improve document layout
Formalization
Decrease number of *layout issues*
Method
Paraphrase

Paraphrase

He says his departure has nothing to do with the resignation calls.



He says his departure is unconnected with the resignation calls.

- = Differing textual realizations of same meaning [Ganitkevitch et al., 2013]
- Wide application [Knight and Marcu, 2000; Siddharthan, 2002]
- Many new methods proposed

  Bilingual pivoting [Zhou et al., 2006], Recursive autoencoder [Socher et al., 2011], Phrase-based monolingual machine translation [Wubben et al., 2012], Convolutional neural network [Yin and Schütze, 2015], …

Idea: Shortening by Paraphrasing

It is not clear what fraction f_{cl} of HVCs with $N(\mathrm{H\,I})$ above the 10^{18} cm^{-2} detection thresh-old will give rise to $W(\mathrm{Mg\,II}) \geq 0.3$ Å because the equivalent

Issue to be fixed
(hyphenated word)

Idea: Shortening by Paraphrasing

It is unclear what fraction f_{cl} of HVCs with $N(HI)$ above the 10^{18} cm^{-2} detection threshold will give rise to $W(MgII) \geq 0.3$ Å because the equivalent

Preceding phrases were **shortened** by paraphrasing

➡ Layout issue was avoided

J. C. Charlton et al. "QSO Absorption Line Constraints on Intragroup High-Velocity Clouds"

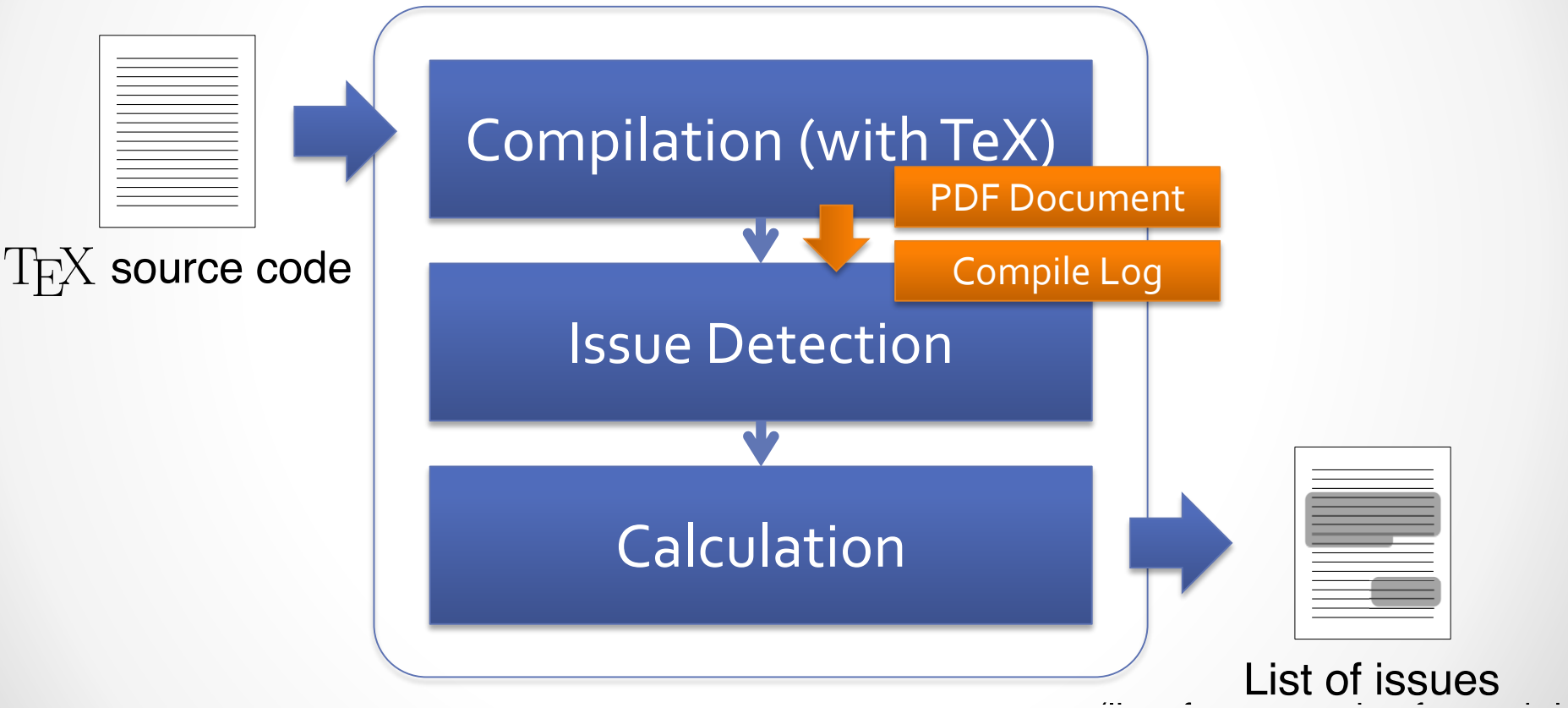
Workflow

1. Issue Detection
2. Candidate Generation
3. Filtering
4. Selection

Workflow

1. Issue Detection
2. Candidate Generation
3. Filtering
4. Selection

1. Issue Detection



- **Layout issues** in PDF document can be detected by using **bounding box information** extracted from PDF file and TeX's **compile log**
- Map issue positions in pages to those in source code file
  ➜ Sequence alignment problem

tion

TEX source code

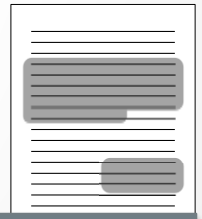
X)

DF Document

Compile Log

Issue Detection

Calculation



In this study, we deal with two kinds of layout issues **widows** and **hyphenations**

es

tion

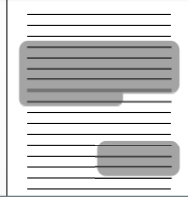
- **Layout issues** in PDF document can be detected by using **bounding box information** extracted from PDF file and TeX's **compile log**
- Map issue positions in pages to those in source code file
  ➜ Sequence alignment p

TEX source code

Issu

For each layout issue, we find:
- Position
- *Target region*
- Required amount of modification

Calculation

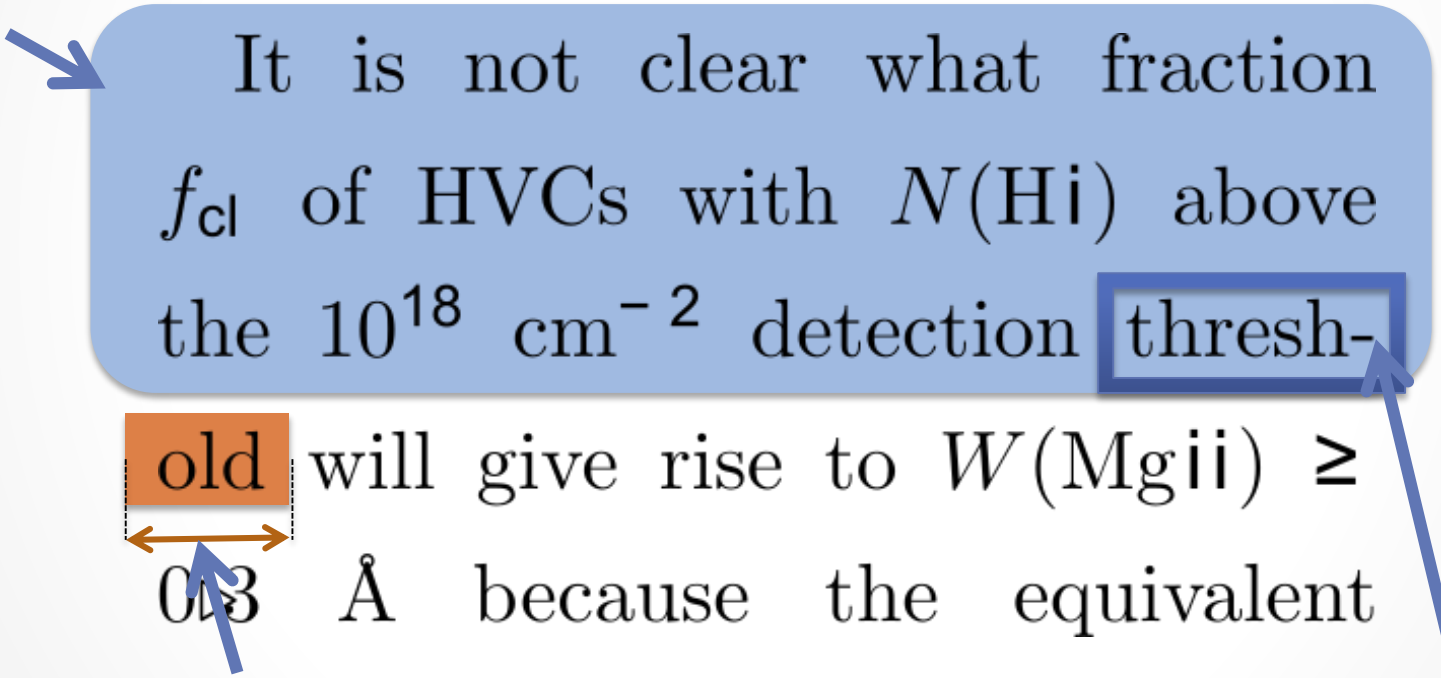


In this study, we deal with two kinds of layout issues **widows** and **hyphenations**

es

1. Issue Detection

Target region



Amount of required modification

Issue to be fixed
(hyphenated word)

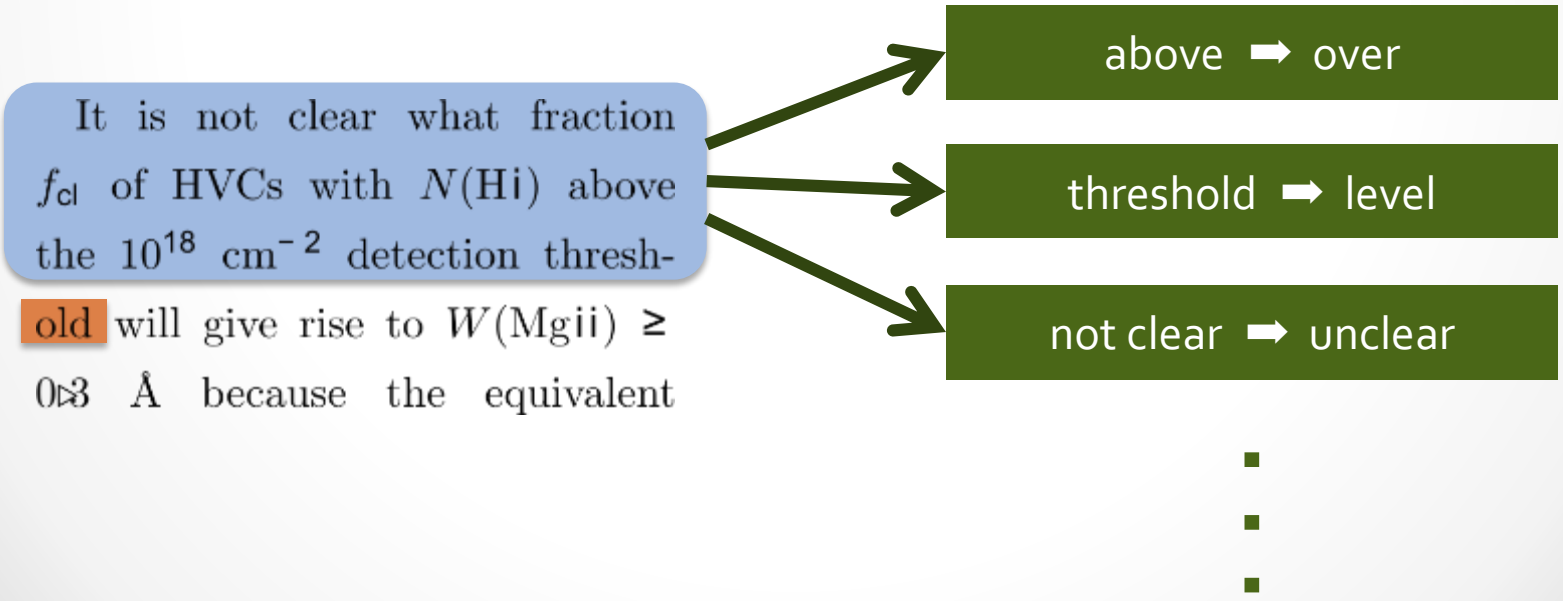
Workflow

1. Issue Detection
2. Candidate Generation
3. Filtering
4. Selection

2. Candidate Generation

2. Candidate Generation

- Generate paraphrases within target region
- Method is simple: **string substitution** by looking up **dictionaries** (language resources, or corpora)



2. Candidate Generation

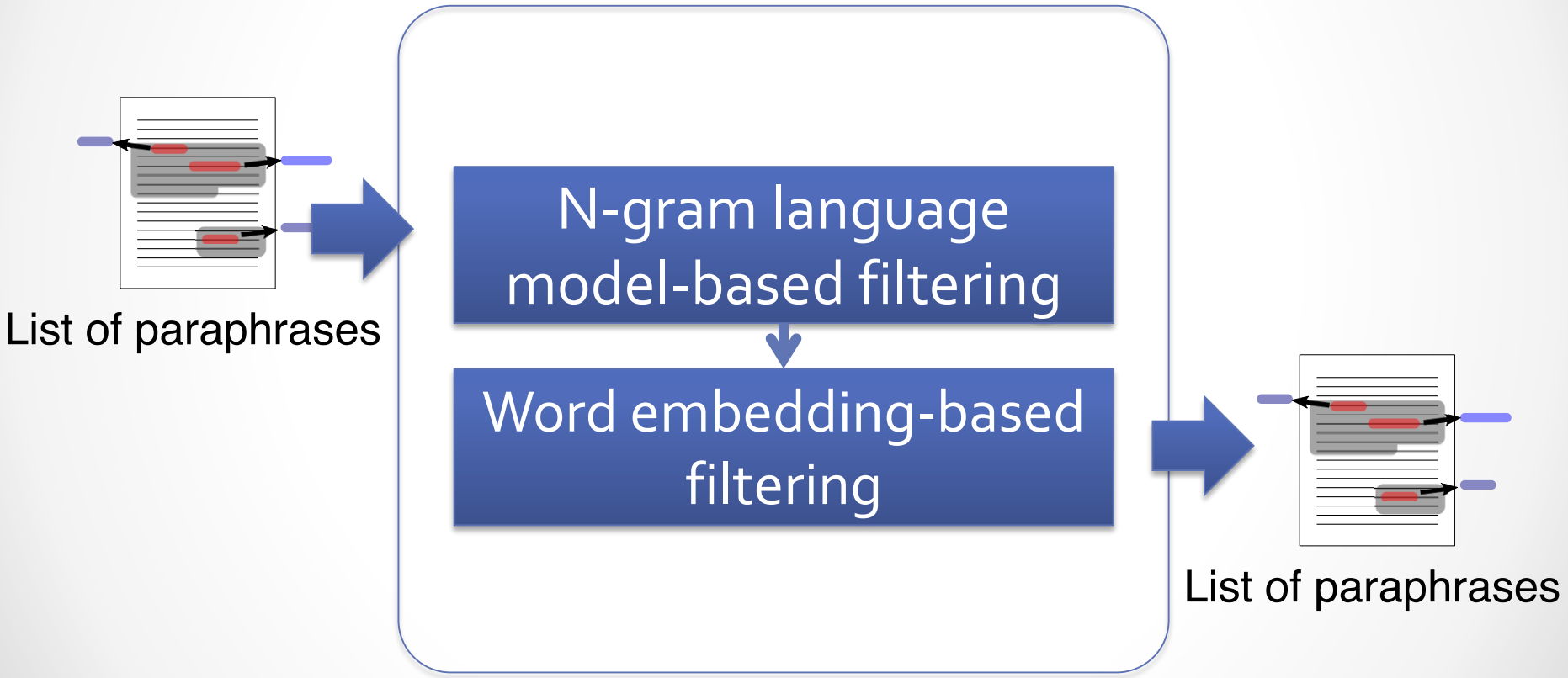
- (Subset of) PPDB 1.0 [Ganitkevitch et al., 2013]
  - List of **synonym** pairs (e.g. {"closely", "nearly"})
  - Automatically generated from large parallel corpora
  - 66,557 entries
- **Phrase** substitution patterns (original work)
  - Collected through analysis of editing history of Wikipedia
  - 142 entries

Filtered only "small" edits, not additions, spams, …, using meta data [Yatskar et al., 2010] and Levenshtein distance between revisions

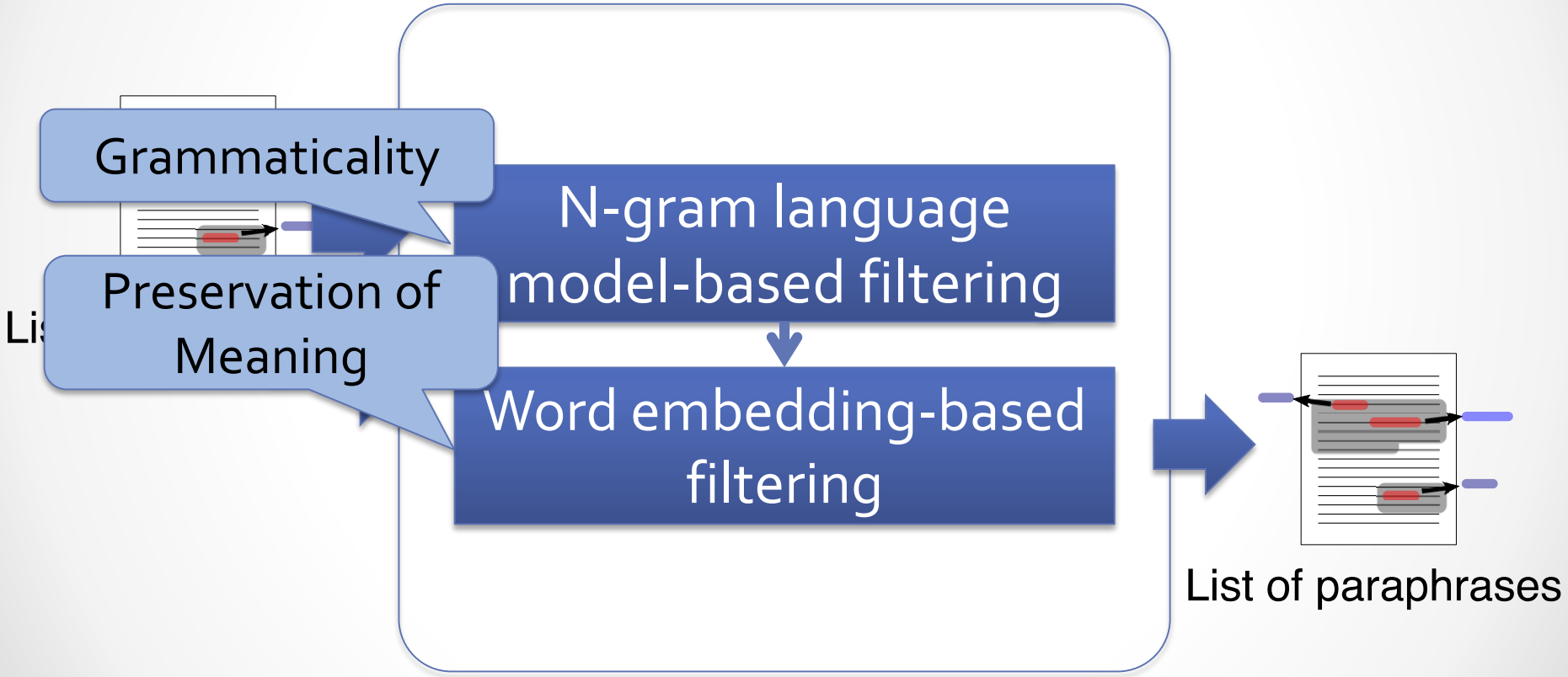
Workflow

1. Issue Detection
2. Candidate Generation
3. Filtering
4. Selection

3. Filtering


List of paraphrases
N-gram language model-based filtering
Word embedding-based filtering
List of paraphrases

3. Filtering


Grammaticality
Preservation of Meaning
N-gram language model-based filtering
Word embedding-based filtering
List of paraphrases

3. Filtering

Not all candidates are appropriate for our purpose

- Every PPDB entiry has score but its quality is unclear [Wieting et al., 2015]
  e.g. {"paid", "paying"} gains high score
- Several *styles* of paraphrases included [Pavlick et al., 2015]
  e.g. {"close", "open"} … really paraphrase?

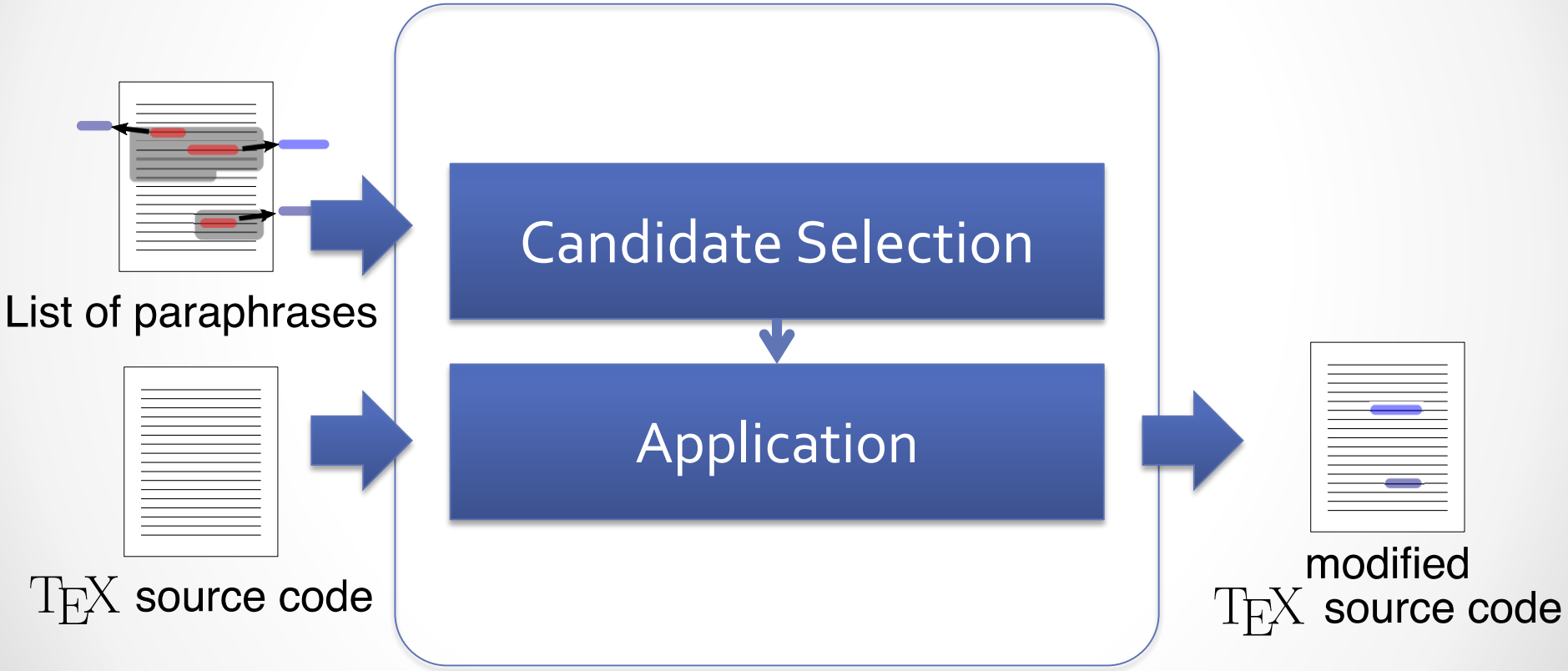
➜ 2 filtering methods based on **n-gram language model** (for **grammaticality**) and **word embedding** (for **preservation of meaning**)

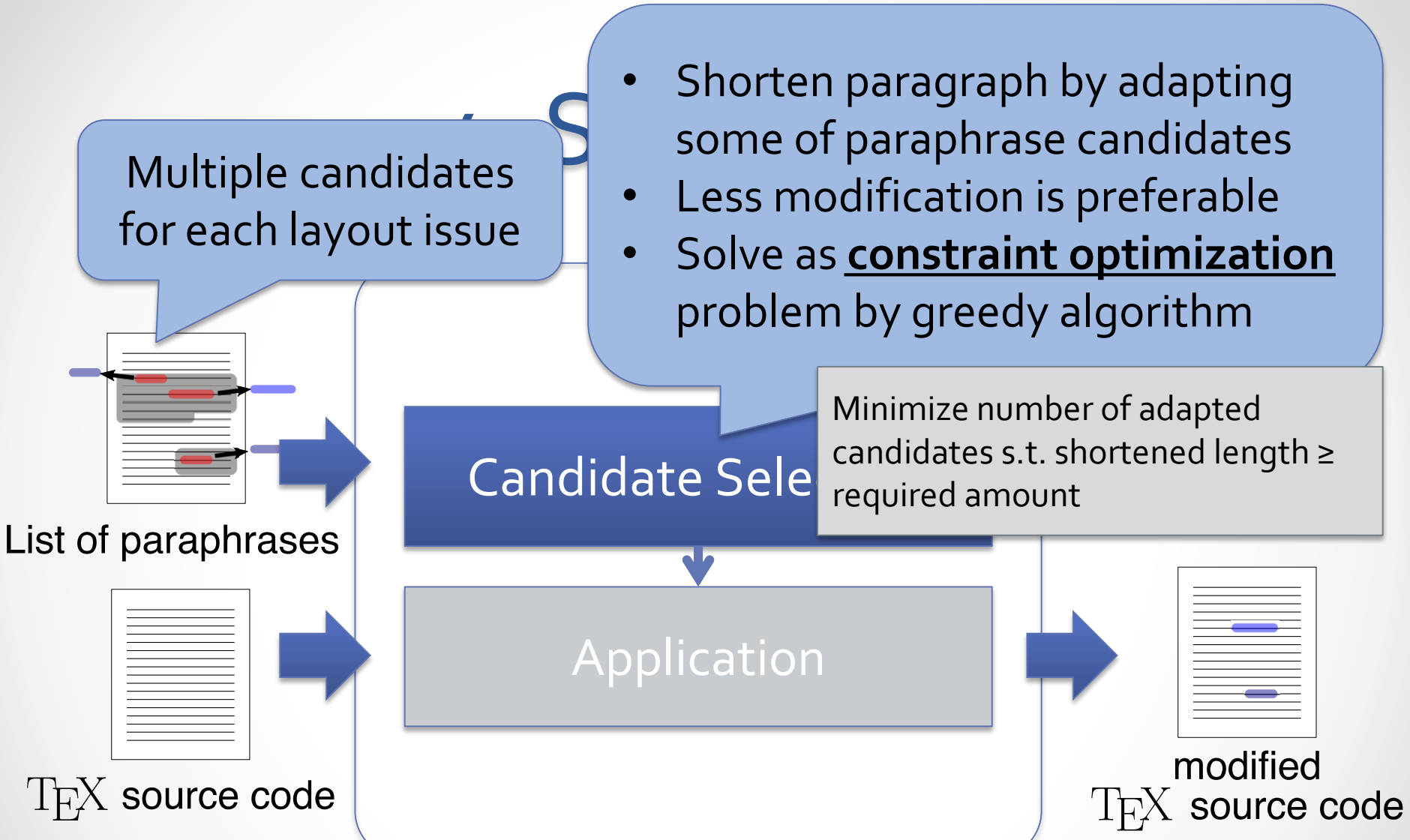
Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. "From Paraphrase Database to Compositional Paraphrase Model and Back." Transactions of the Association for Computational Linguistics, vol. 3, pp. 345-358, 2015.
Pavlick, E., Bos, J., Nissim, M., Beller, C., Van Durme, B. and Callison-Burch, C. "Adding Semantics to Data-Driven Paraphrasing." In the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. 2015.

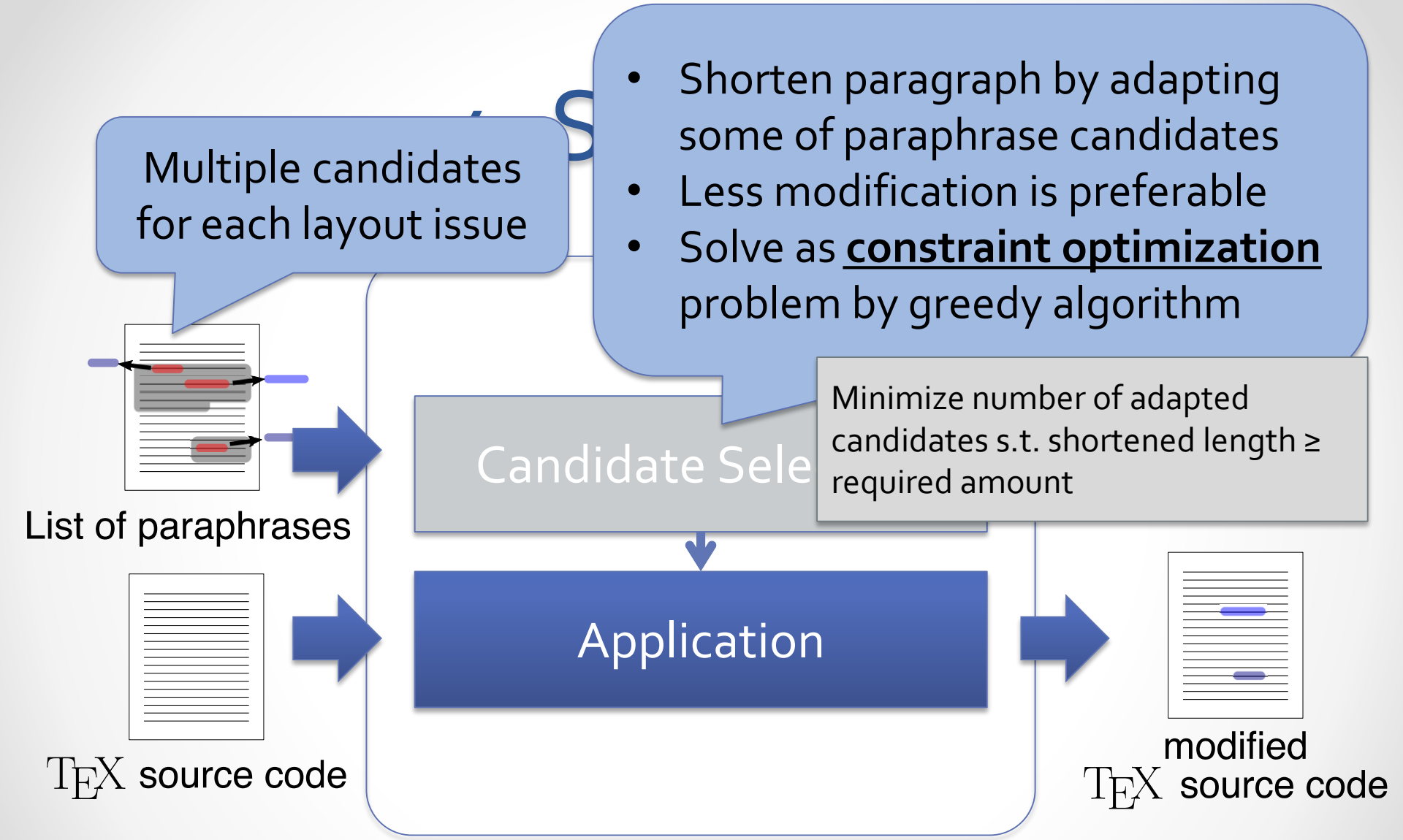
Workflow

1. Issue Detection
2. Candidate Generation
3. Filtering
4. Selection

4. Selection


List of paraphrases
TEX source code
Candidate Selection
Application
modified
TEX source code


Multiple candidates for each layout issue
• Shorten paragraph by adapting some of paraphrase candidates
• Less modification is preferable
• Solve as constraint optimization problem by greedy algorithm
Minimize number of adapted candidates s.t. shortened length ≥ required amount
Candidate Sele
List of paraphrases
Application
TEX source code
modified TEX source code

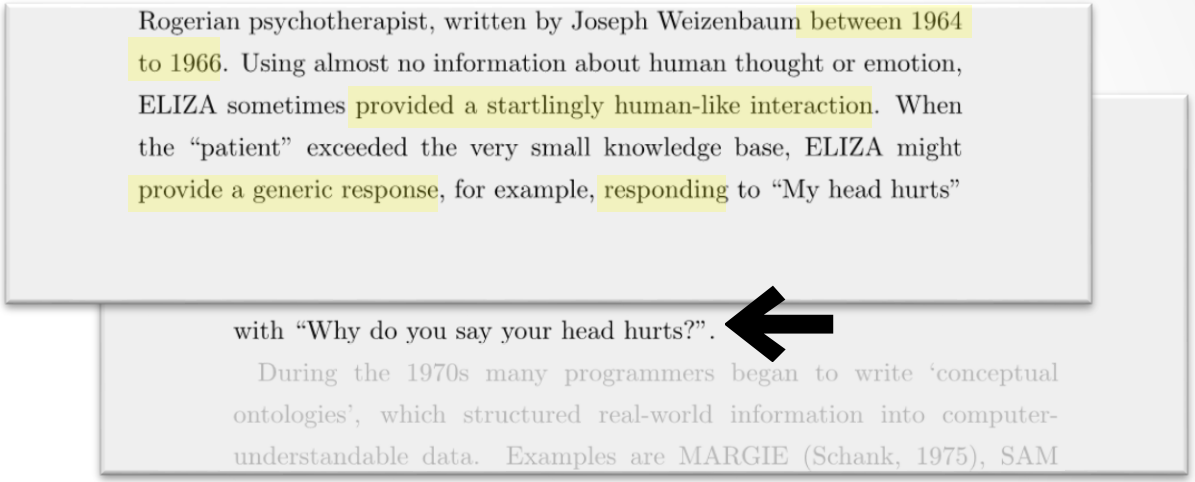

Multiple candidates for each layout issue
• Shorten paragraph by adapting some of paraphrase candidates
• Less modification is preferable
• Solve as constraint optimization problem by greedy algorithm
Minimize number of adapted candidates s.t. shortened length ≥ required amount
List of paraphrases
Candidate Sele
Application
TEX source code
modified TEX source code

Experiment

- Apply our method to 16 test documents
- Compare numbers of issues before/after applying
- Test documents
  - Collect plain English texts from the Web (arXiv, English Wikipedia, Project Gutenberg)
  - Generate pseudo-scientific documents by applying TeX template to them

Results (1/3)

Before


Rogerian psychotherapist, written by Joseph Weizenbaum between 1964 to 1966. Using almost no information about human thought or emotion, ELIZA sometimes provided a startlingly human-like interaction. When the "patient" exceeded the very small knowledge base, ELIZA might provide a generic response, for example, responding to "My head hurts"

with "Why do you say your head hurts?".

During the 1970s many programmers began to write 'conceptual ontologies', which structured real-world information into computer-understandable data. Examples are MARGIE (Schank, 1975), SAM

- 1964 ➡ '64
- provided (v. p.) ➡ gave (v. p.)
- response (n. sing.) ➡ reply (n. sing.)
- responding (v. gerund) ➡ replying (v. gerund)

:

Results (1/3)

Before

Rogerian psychotherapist, written by Joseph Weizenbaum between 1964 to 1966. Using almost no information about human thought or emotion, ELIZA sometimes provided a startlingly human-like interaction. When the "patient" exceeded the very small knowledge base, ELIZA might provide a generic response, for example, responding to "My head hurts" with "Why do you say your head hurts?".

During the 1970s many programmers began to write 'conceptual ontologies', which structured real-world information into computer-understandable data. Examples are MARGIE (Schank, 1975), SAM

After

apist, written by Joseph Weizenbaum between '64 to '66. Using almost no information about human thought or emotion, ELIZA sometimes gave a startlingly human-like interact. When the "patient" exceeded the very small knowledge base, ELIZA might give a generic reply, for example, replying to "My head hurts" with "Why do you say your head hurts?".

Results (2/3)

	Layout	Content	# pp.	Hyphenations	Widows
1	Double-column	Gutenberg	7	**31 → 26 (+6, -11)**	3 → 3 (+0, -0)
2		arXiv	7	**67 → 58 (+26, -35)**	**1 → 0 (+0, -1)**
3		Gutenberg	8	**47 → 45 (+10, -12)**	**3 → 1 (+0, -2)**
4		arXiv	9	**97 → 92 (+52, -47)**	**2 → 1 (+0, -1)**
5		Wikipedia	11	**201 → 151 (+56, -106)**	0 → 0 (+0, -0)
6		Wikipedia	11	**194 → 174 (+50, -70)**	**2 → 1 (+0, -1)**
7		Wikipedia	11	**144 → 120 (+25, -49)**	**3 → 1 (+1, -3)**
8		Gutenberg	14	**113 → 88 (+26, -51)**	4 → 4 (+4, -4)

Results (3/3)

	Layout	Content	# pp.	Hyphenations	Widows
9	Single-column	Gutenberg	7	**14 → 11 (+4, -7)**	**2 → 0 (+0, -2)**
10		Wikipedia	7	**19 → 17 (+7, -9)**	1 → 1 (+0, -1)
11		arXiv	8	**16 → 10 (+7, -13)**	**1 → 0 (+0, -1)**
12		Gutenberg	9	**15 → 6 (+1, -10)**	3 → 3 (+0, -0)
13		Gutenberg	10	11 → 11 (+0, -0)	**3 → 0 (+0, -3)**
14		arXiv	10	**22 → 18 (+11, -15)**	**1 → 0 (+0, -1)**
15		Wikipedia	11	**39 → 26 (+4, -17)**	**5 → 1 (+0, -4)**
16		Wikipedia	12	**49 → 36 (+20, -33)**	**3 → 0 (+0, -3)**

Discussion

- Effective to hyphenations, not so much to widows
  - Only simple paraphrasing methods
  - Cannot generate large amount of modification
    ➜ Treated only shortening, but **lengthening** would also help
- Quality of paraphrases
  - Need resources that are more suitable for our purpose
  - Quantitative evaluation is difficult
    ➜ Leave judgement to **human user**
    (System just suggests candidates as assistance)
- Used TeX as black box

  ➜ More understanding of behavior of TeX

Summary

- Applying NLP techniques like paraphrasing to document layout optimization helps computer-based system do jobs that only humans could do
- Our result is still preliminary, but suggests that this kind of application is promising and leaves room for research