Improving **Automated Controversy Detection** on the Web

Myungha Jang and James Allan

College of Information and Computer Sciences, Center for Intelligent Information Retrieval,
University of Massachusetts Amherst

www.myungha.com & mhjang@cs.umass.edu

Motivation

In the age of "Filter Bubble", readers who have not heard of the controversy is likely to be misled or uninformed.
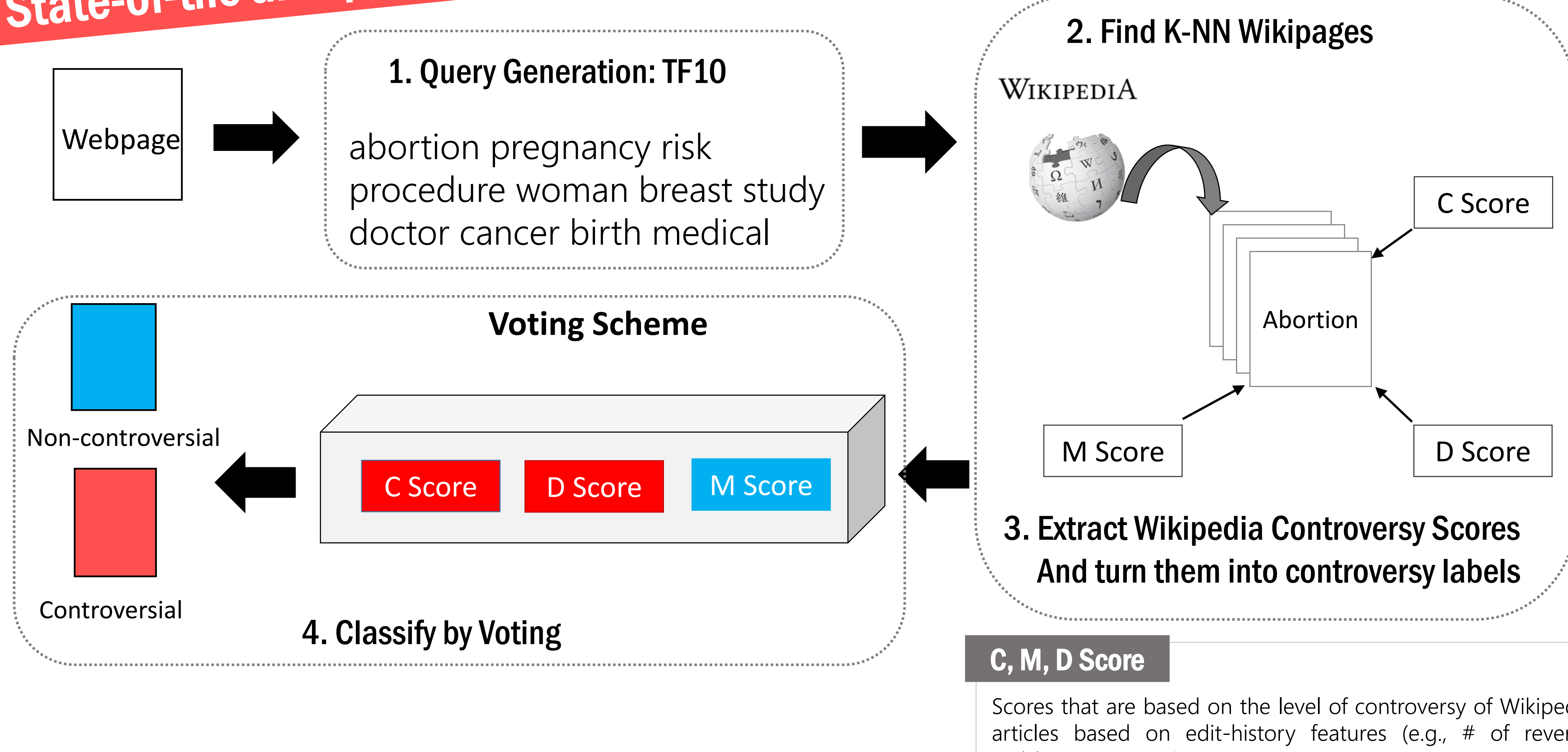
If Google Was A Guy (Part 3)

- Well, I have one million results that say they don't, and one result that says they do

(Part 3)
- Vaccines cause autism

- I knew it!

Watch the full Youtube Clip
"If Google Was A Guy"

Research Goal

Automatically warning users when a Web-page discusses controversial topics

Dear Britain, if you stay in the EU, you will ruin our lives. Here is why
Jean Quatremer

Only a vote for Brexit can save the European Union. So let yourself be won over by such brilliant leaders as Nigel Farage and Boris Johnson

Given the facts the UK should stay in the EU

While both pro- and anti-EU campaigns exaggerate their case, the facts suggest saying no to Brexit would be the better option ... and the EU

Controversial

State-of-the art Approach

Webpage →

1. Query Generation: TF10

abortion pregnancy risk procedure woman breast study doctor cancer birth medical

2. Find K-NN Wikipages

WIKIPEDIA

Abortion

C Score
M Score
D Score

3. Extract Wikipedia Controversy Scores And turn them into controversy labels

Voting Scheme

Non-controversial

| C Score | D Score | M Score |

Controversial

4. Classify by Voting

C, M, D Score
Scores that are based on the level of controversy of Wikipedia articles based on edit-history features (e.g., # of reverts, revision counts, etc)

S. Dori-Hacohen and J. Allan , Automated Controversy Detection on the Web, ECIR 2015

Limitations ☹

1. Ambiguous Query Generation

The generated query ends up containing an unknown mixture of different sub-topics of a document.

➡ **Solution 1: Generating multiple queries from smaller but more coherent blocks (or tiles) of the document**

2. Underrated Wikipedia Controversy Scores

The controversy level is underestimated on specific and sub-topical Wikipages whose topical disputes have often been delegated by other Wikipages on the broader topic.

➡ **Solution 2. Smoothing Wikipedia controversy scores using neighbors**

Solution 1: Finding KNN Pages with TileQuery

1. Document Segmentation

Split a document into multiple block of sentences (tiles) using *TextTiling*.

Webpage →

2. Query Generation

Generate a query of a mixture of g (global) and l (local) most frequent terms from the tile.

→ Ranked list

3. Aggregating the Ranked Lists

Compute the relevance of each retrieved Wikipedia page w_i by aggregating its rank scores as follows:

$$Relevance(w_i) = \sum_{t_i \in T} (K - rank_{t_i}(w_i))$$

Ranked list → Ranked list

t Ranked lists

Solution 2: Smoothing Wikipedia Controversy Scores

1. Construct Wikipedia Topic Hierarchy using their titles' prefix-relation

Clique-based   Pair-based

abortion

abortion_act   abortion_rights   abortion_in_the_us

abortion_rights_violence

abortion_act_1967   abortion_rights_movement   abortion_in_the_us_state_by_state

2. Network-based Smoothing

Abortion
Abortion_Act
Abortion in_the_US
Abortion rights
Abortion in England
Abortion in China

$$C'(W_i) = \sum_{W_j \in inLinks(W_i)} \frac{C(W_j) * r_j}{\sum r_k}$$

where r_i is a revision count of W_i.

	Original scores		Revised scores	
	M	C	M	C
Abortion	4,102,593	0.300	4,102,593	0.300
Abortion_Act	0	0		
Abortion_in_China	0	0	2,062,156	0.166
Abortion_in_England	0	0	2,128,909	0.172
Abortion_in_the_US	0	0.002	2,983,300	0.218

Two controversy scores on "abortion"-related pages, before and after smoothing the score
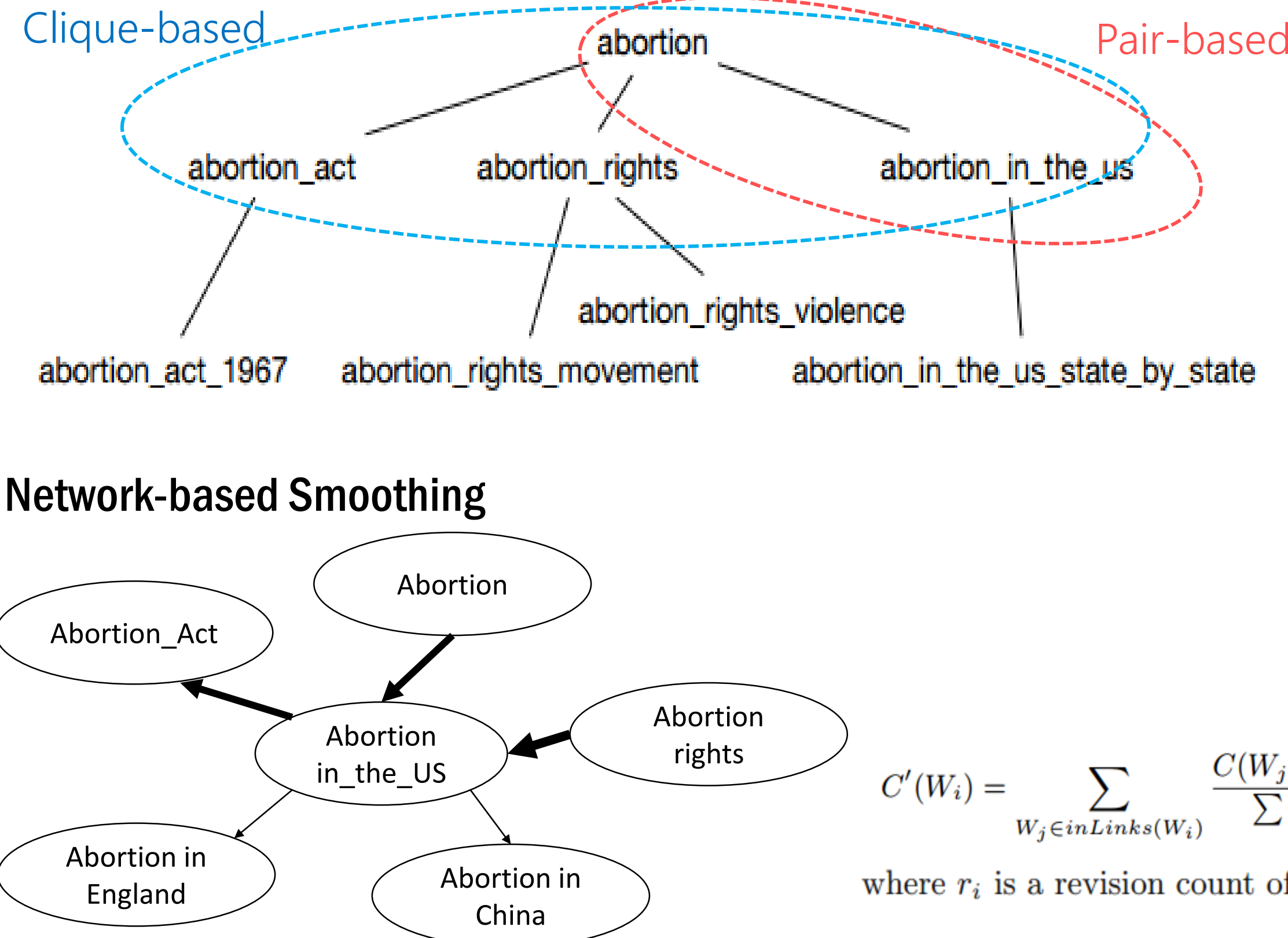
Experiments

Dataset
- 303 Web documents
- Collected from topics with varying controversy level
- 42%: controversial

Extensive Parameter Explorations:
- Query Methods = {AllQuery, TF10, TileQuery}
- # of Wikipedia Neighbors, K = {1... 20}
- Use of score smoothing
- Thresholds for C and M Scores
- Wikipedia neighbors selection = {Pair, Clique}
- Voting Method = {M, C, D, Majority, Or/AND}

Table 2: Accuracy, F1, and the best parameters in 5-fold runs for different query and inferred score settings.

Run	Query	Inferred Score	K	C Threshold	M Threshold	Aggregation	Acc.	F1	
1		N/A	{5, 20}	{0.17, 4.18 · 10⁻²}	{40000,20000}	{M, Maj.}	0.72	0.50	
2	ALL	Clique	15	{0.17, 4.18 · 10⁻²}	{40000,20000}	{M, Maj.}	0.78	0.68	
3		Pair	{5, 20}	{0.17, 4.18 · 10⁻²}	{40000,20000}	{M, Maj.}	0.73	0.53	
4		N/A	20	4.18 · 10⁻²	{20000, 40000, 84930}	{M, Maj.}	0.75	0.57	State-of-the-art
5	TF10	Clique	20	4.18 · 10⁻²	84930	Maj.	0.79	0.68	
6		Pair	{10, 20}	4.18 · 10⁻²	{20000, 84930}	Maj.	0.75	0.57	
7		N/A	{10,15,20}	4.18 · 10⁻²	{40000,20000}	{M, Maj.}	0.75	0.59	
8	TILE	Clique	20	0.17	40000	M	0.80	0.71	Our Best Run
9		Pair	{10,15,20}	4.18 · 10⁻²	{40000,20000}	{M, Maj.}	0.75	0.61	

$$Relevance(w_i) = \sum_{t_i \in T} (K - rank_{t_i}(w_i))$$

Conclusion and Future Work

- Our modifications of TileQuery and Wikipedia controversy score smoothing improved the state-of-the-art controversy detection by 5% in Acc and 14% in F1.
- Title prefix-relation only covered 5~10% of the Wikipedia titles. More sophisticated methods to find Wikipedia neighbors for smoothing can be studied.
- We plan to further investigate different scenarios where TF10 and TileQuery works well.