Statute Recommendation:
Re-ranking Statutes by Modeling Case-Statute Relation with Interpretable Hand-crafted Features

Chuanyi Li[1], Jidong Ge[1,*], Kun Chen[1], Bin Luo[1], and Victor Chang[2,*]

1. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

2. Department of Operations and Information Management, Aston Business School, Aston University, Birmingham, UK

Contents

Background: AI and Law

- The increasing needs of legal guidance in divorce cases.

- Values of an intelligent **Statute(Law Article) Recommending** tool:
  - Help lawyers and judges handle divorce cases properly.
  - Help lawyers and judges improve work efficiency.
  - Help common people derive advices on defending their rights.

- One of the mostly concerned aspects of AI and Law research.
  - **Legal judgment prediction**; Legal cases retrieval; Court view generation, etc.

Law article prediction | Charge prediction | Prison Term prediction

Related Techniques of Law Article Prediction

- Recommender Systems
  - One common way to improve recommending effectiveness is to utilize novel features of the recommended items and the input query.
  - None of the existing LAP approaches utilized **case-statute relationship** explicitly.
- Collaborative Filtering
  - Widely used in the recommender systems
  - None of the existing LAP approaches discuss **how to retrieve statutes candidates** in detail.
- Learn to Rank
  - Widely used in information retrieval and recommendation
  - We are the first adopting Learn to Rank techniques in LAP.

Approach: Re-ranking Framework

- Derive statute candidates by retrieving similar cases
  - Calculate similarity between legal cases: using keywords
  - Considering both efficiency and accuracy
- Re-rank statute candidates
  - Define the Case-Statute relational features
  - Apply the features in different Rankers

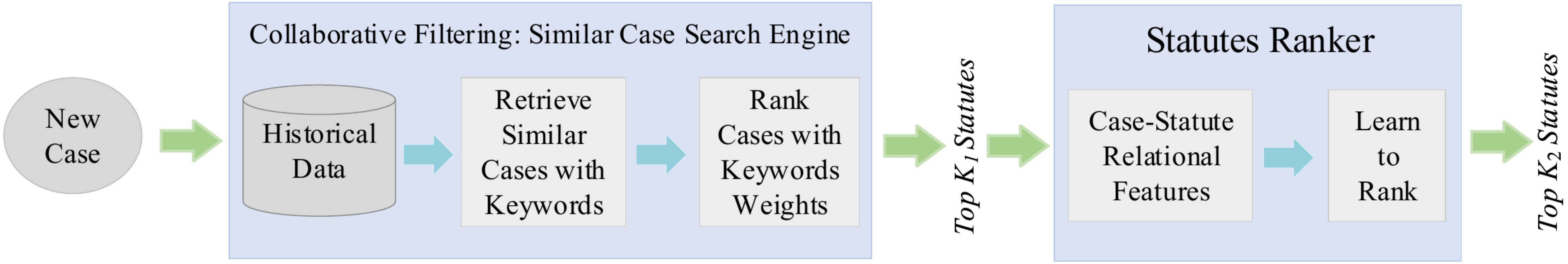


Figure 1: Framework of the proposed statute recommending approach.

Approach: Real-world Corpus

- The existing public datasets like CAIL2018 are not applicable for the proposed approach.
  - No original legal judgment document supporting retrieving required fields

# of Case	Maxlen of Case	Minlen of Case	Avg.len of Case	# of Statutes	Max Citation	Min Citation	Avg. Citation
19860	4889	24	347.6	456	16	1	4.01

- Chinese Judgment Online
- 2015~2016 divorce cases
- First-round civil instances
- 19860 cases

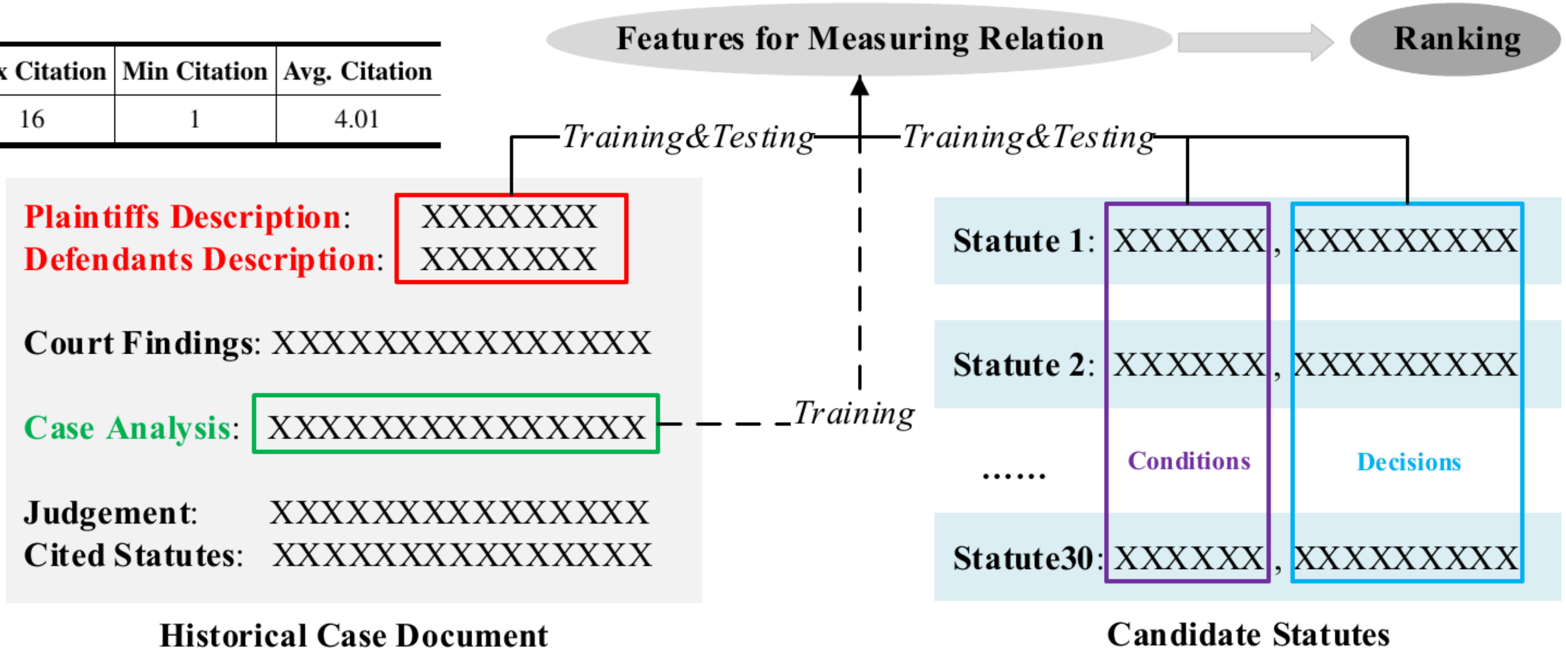


Figure 2: Case Document, Statutes Content and the relational features

Approach: Similar cases searching (1)

- Steps
  - Segmentation, TF-IDF, and Keywords [Tried methods: **IG, WLLR, WFO**]:

    Tfidf_IG **Tfidf_WLLR** **Tfidf_WFO**

    - $Case = \{Word | Word = (word, POS, TFIDF, W_{key})\}$
  - Inverse Index
    - $Word = \{(C_{ID1}, POS_1, TFIDF_1, W_{key1}), \ldots\ldots, (C_{IDn}, POS_n, TFIDF_n, W_{keyn})\}$
  - Weights of cases

$$\boxed{Weight(C) = \alpha \sum_{i=1}^{m} f(POS_i) + \sum_{i=1}^{m} TFIDF_i + \beta \sum_{i=1}^{m} W_{keyi} \qquad f(POS) = \begin{cases} w_s, & \text{if } POS \in \mathbb{S}; \\ w_o, & \text{if } POS \notin \mathbb{S}. \end{cases}}$$

$C_{new} \cap C = \{Word_1, Word_2, \ldots, Word_m\}$

$POS_i, TFIDF_i, W_{keyi}$ are corresponding properties of $Word_i$

$Special$ tag in the set $\mathbb{S}$.

  - Weights of statutes

$$\boxed{StatuteWeight(S) = \sum_{i=1}^{l} Weight(C_i)}$$

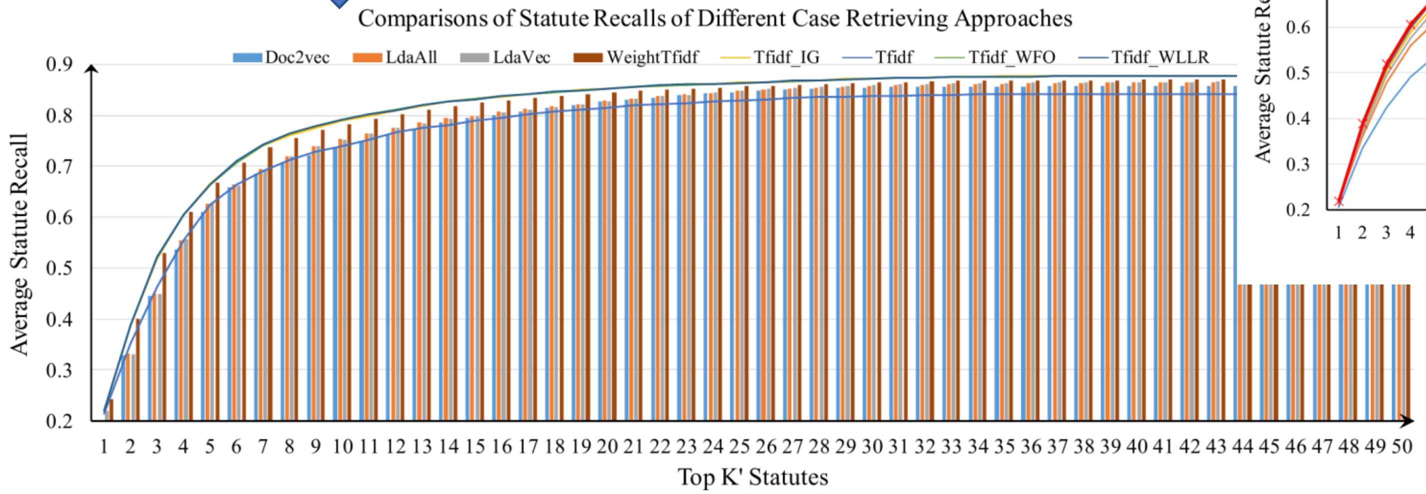
- Baselines:
  - Tfidf; WeightTfidf; LDA-related; Doc2Vec

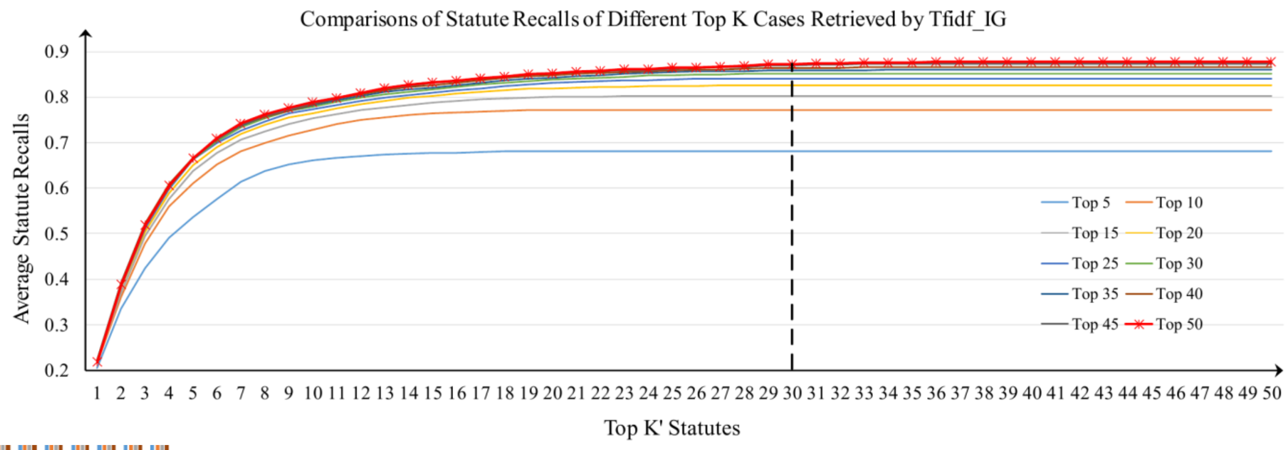
Approach: Similar cases searching (2)

- Results
  - Evaluation Metrics: $Recall@i = \frac{tp_i}{g}, Precision@i = \frac{tp_i}{i}, AP@i = \frac{1}{i}\sum_{j=1}^{i} Precision@j \times Cited@j, where$

    $$Cited@i = \begin{cases} 0, & \text{if } statute@i \text{ is not cited by the case;} \\ 1, & \text{if } statute@i \text{ is cited by the case.} \end{cases}$$

  - Retrieve Top K cases for Top K' Statutes:
    - The proper method?
    - The proper K and K'?



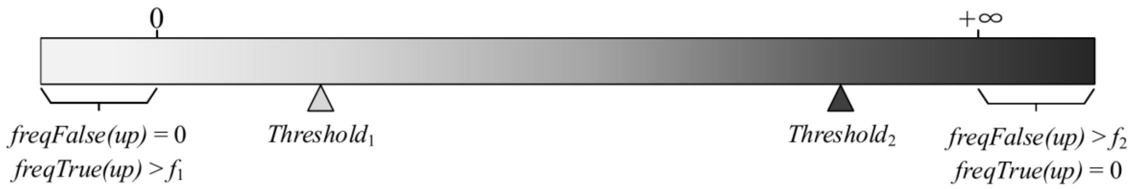


We set K=50, K'=30.

Approach: Modeling Case-Statute Relation

- Basic Features
  - Initial Unigram Pairs (UP) **➔1.5 million UPs**
    - One UP consists of a word from the case and a word from the statute ➔ The size of UPs is too large to train a ranker.
  - Selected Unigram Pairs **➔3967UPs**
    - f_1=5,f_2=200,$Threshold_1$=1,$Threshold_2$=100

$$\forall\ up \in UP_{CiteFalse} \cup UP_{CiteTrue}\text{, the scope of } \frac{freqFalse(up)}{freqTrue(up)} \text{ is:}$$



- Novel Features: hand-crafted features
  - Words Overlap
  - Roles and Special Group Consistency
  - Judgment Consistency
  - Relationship Consistency
  - Features based on LDA

Are factors often considered by judges and lawyers when they handle divorce cases

For each candidate statute of the input legal case, the values of each feature would be calculated according to the definitions of each feature. Then they would be applied to the re-ranking model for ranking candidate statutes.

Approach: Training Rankers

- Point-wise ranking-only considering the <query, item>
  - Classification or Regression
  - Treat case-statute suitableness as 0/1 classification problem
    - We use **BERT**
- Pair-wise ranking-considering the <item1,item2> pairs of a query
  - We use **SVMRank**
- List-wise ranking-considering the {<query, items>} list
  - We use **DLCM**(Deep Listwise Context Model)

- For different Learn to rank algorithm, we try different features

[CLS] vector of BERT	**[CLS] vector of LBERT**	**Basic Features**	**Novel Features**

Experiments and Evaluation: RQ1 (1)

- **RQ1**: How much can the proposed relational features improve LAP performance comparing with the others?

- Searching Engine: purely rely on CF
- C-Multilabel-BERT: treat statutes as labels

- R-SVM-Basic+Novel achieves SOTA
- **However, there is still a big gap between SOTA and Golden Standard.**

Approaches	Recalls (%)				Precisions (%)			
	@1	@3	@5	@10	@1	@3	@5	@10
Search Engine	21.86	51.84	66.56	78.89	59.25	48.58	37.80	22.80
C-MultiLabel-BERT	18.38	42.32	62.73	77.05	47.53	40.54	34.82	22.15
C-MultiLabel-BERT-Tuning	17.72	42.46	62.88	77.26	47.31	40.55	34.83	22.17
C-MultiLabel-LBERT	17.26	42.43	62.67	77.13	47.25	40.55	34.82	22.16
C-MultiLabel-LBERT-Tuning	17.95	42.61	62.84	77.22	47.48	40.56	34.83	22.17
C-Binary-BERT	22.12	51.04	64.43	73.82	60.04	48.35	37.03	21.14
C-Binary-BERT-Tuning	22.86	52.41	66.25	74.63	60.63	49.14	38.35	21.53
C-Binary-LBERT	23.26	51.47	65.83	74.60	61.87	48.37	37.62	21.53
C-Binary-LBERT-Tuning	23.53	53.00	66.41	74.96	62.01	51.04	38.11	21.67
DLCM-BERT	26.05	54.25	67.81	79.74	73.42	53.17	38.45	23.52
DLCM-BERT-Tuning	26.11	54.28	67.80	79.45	73.44	53.18	38.45	23.51
DLCM-LBERT	26.10	54.22	67.80	79.46	73.44	53.16	38.31	23.50
DLCM-LBERT-Tuning	26.12	54.64	67.85	79.76	73.45	53.17	38.49	23.54
DLCM-Basic	23.70	50.45	65.55	79.17	62.16	47.02	37.57	22.96
DLCM-Basic+Novel	23.71	52.01	66.89	78.84	62.20	48.38	38.12	22.83
R-SVM-BERT	23.79	52.56	66.58	79.04	64.74	49.30	37.89	22.94
R-SVM-LBERT	23.81	53.12	66.23	79.16	65.52	51.74	37.75	22.98
R-SVM-Basic	25.46	54.22	67.18	79.42	70.38	52.90	38.22	23.03
R-SVM-Basic+Novel	**26.52**	**55.43**	**68.14**	**79.94**	**74.83**	**55.03**	**38.59**	**23.67**
Gold Standard	*38.13*	*82.38*	*86.70*	*87.14*	*99.98*	*76.98*	*50.56*	*25.66*

Experiments and Evaluation: RQ1 (2)

- **RQ1**: How much can the proposed relational features improve LAP performance comparing with the others?

- MAP values:
    - Figure 6 shows the highest MAP is @3
    - Average number of cited statutes by a case is 4. Theoretically, the highest MAP should be achieved @4.
- Balance the 0/1 labels

	Balanced Strategy	Recalls (%) @1	Recalls (%) @3	Recalls (%) @5	Recalls (%) @10	Precisions (%) @1	Precisions (%) @3	Precisions (%) @5	Precisions (%) @10
C-Binary-BERT	Unbalanced	22.12	51.04	64.43	73.82	60.04	48.35	37.03	21.14
	Simple	22.84	50.73	64.80	77.22	61.31	47.99	37.09	22.19
	SF-ICF	23.55	51.56	65.51	78.13	62.00	48.40	37.45	22.57

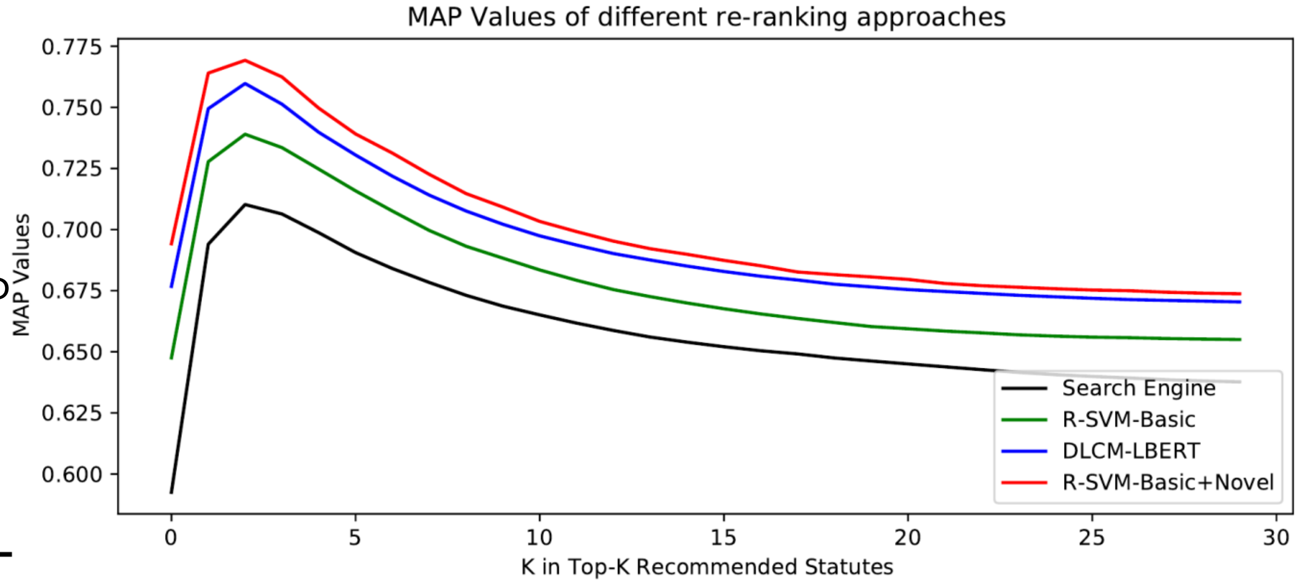


Figure 6: MAP@Top K of four different re-ranking approaches.

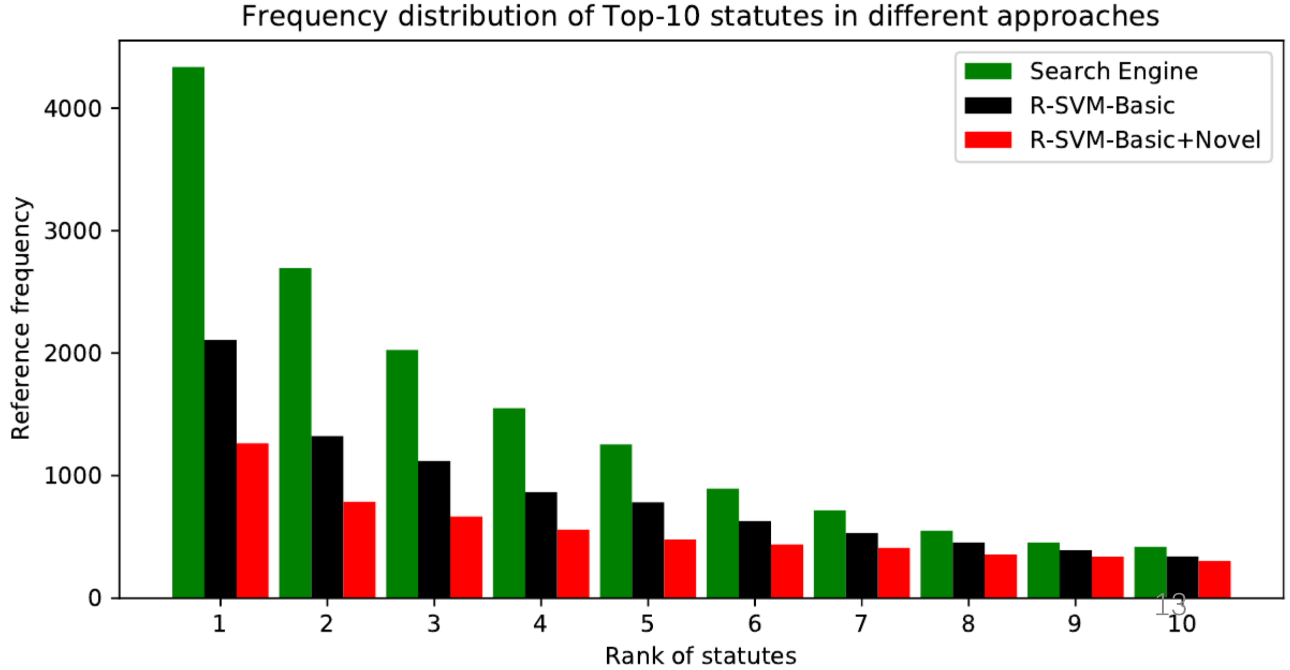
Experiments and Evaluation: RQ2

- **RQ2**: To what extent do the novel relational features contribute to the re-ranking approach?
  - Basic: **3967** dimensional vector; Novel: **441** dimensional vector.
  - Recall@K and Precision@K
  - MAP@K

R-SVM-Basic	25.46	54.22	67.18	79.42	70.38	52.90	38.22	23.03
R-SVM-Basic+Novel	**26.52**	**55.43**	**68.14**	**79.94**	**74.83**	**55.03**	**38.59**	**23.67**

- Frequently cited statutes in Top_K recommended ones.

Novel features reduce the impacts of the frequency of a statute showing as the Positive instance in the training set.



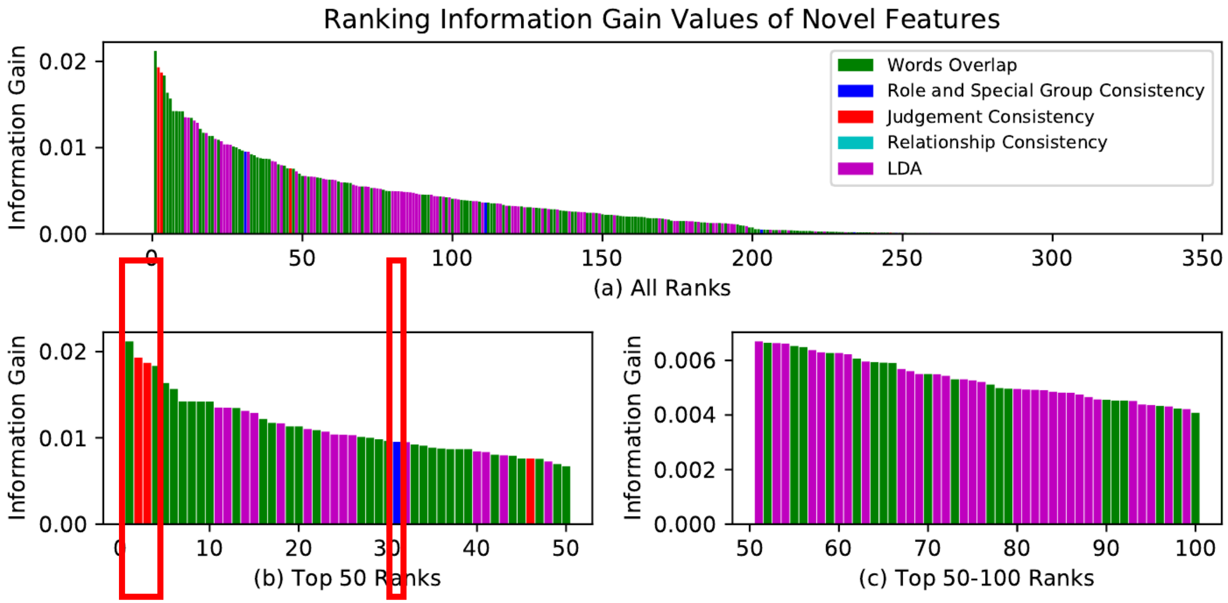
Experiments and Evaluation: RQ3 (1)

- **RQ3**: What kinds of human interpretable information in cases and statutes play a more critical role in ranking statutes than others?

Word overlap and LDA play a more important role in training a better ranker.

Table 5: Feature ablation results by using MAP@1 measuring performance.

Words Overlap	LDA	Role and Special Group Consistency	Judgement Consistency	Relationship Consistency
0.6810	0.6886	0.6893	0.6893	0.6908
0.6791	0.6878	0.6880	0.6904	–
0.6782	0.6867	0.6891	–	–
0.6774	0.6865	–	–	–



several dimensions of **Role** and **Judgement** consistency features

Experiments and Evaluation: RQ3 (2)

We can sort out the guidelines for writing case descriptions and judgment documents that help recommend suitable statutes.

Table 6: Definitions of feature dimensions with high IG values and their indicated instructions.

Feature Type	**Rank of Dimension**	**Definition of the Dimension**	**Indicated Instructions**
Words Overlap	Top 50	Frequently used nouns and verbs in statutes	(1) Invite legal professionals to help write case description. (2) Improve the general public's legal knowledge.
LDA	Top 50	*Shared model based on analysis paragraph*	Explain the relationship between the case description and the applicable statutes in detial in the case analysis section while writing Judgement Documents.
Judgement Consistency	No. 2	Defendant's Attitude	The defendant should express his/her attitude clearly, since it may affect the judge's judgment and thus the suitableness of statutes.
	No. 3	The first time filed for divorce	Whether to file a divorce lawsuit multiple times is critical. Never hide the previous divorce lawsuit nor make up ones in order to win the sympathy of the judge.
	No. 46	"during/after the divorce" in the statute	Describe clearly the performance of the two parties during the divorce and the claims after the divorce.
Role and Special Group Consistency	No. 31	Children of plaintiff/defendant	Describe clearly the situation of children the plan of raising children after divorce.

Experiments and Evaluation: Scalability

- How to handle the increasing scale of
  - historical cases,

1. The TF_IDF vectors of all cases should be reconstructed.
2. The keywords derived through calculating the IG of each word should be re-generated.
3. The search base could not be updated until the number of newly collected cases reached a certain threshold.
4. More cases could be used to train the ranker→ since the case-statute relational features could be easily re-calculated.

  - applicable statutes,

The thing to be updated is to collect the specific content of these statutes to re-calculate the features for training rankers.

  - similar cases, and
  - statute candidates?

The process of building the approach would not be affected.

Conclusion and Future Work

- Similar cases searching: TF_IDF+IG achieves the best results.
- Re-ranking statutes: SVM Rank using Basic and Novel features achieves best results.

- However, there is still a big gap between the proposed approach and the gold standard.

R-SVM-Basic+Novel	**26.52**	**55.43**	**68.14**	**79.94**	**74.83**	**55.03**	**38.59**	**23.67**
Gold Standard	*38.13*	*82.38*	*86.70*	*87.14*	*99.98*	*76.98*	*50.56*	*25.66*

- Future work: define an abstract reasoning model manually and mine a knowledge base for reasoning from case to statute.

Thank you!

lcy@nju.edu.cn