Matching law cases and reference law provision with a neural attention model.

Guoyu Tang
IBM China Research Laboratory
Beijing, China
bjtanggy@cn.ibm.com

Honglei Guo
IBM China Research Laboratory
Beijing, China
guohl@cn.ibm.com

Zhili Guo
IBM China Research Laboratory
Beijing, China
guozhili@cn.ibm.com

Song Xu
IBM China Research Laboratory
Beijing, China
xsxuxs@cn.ibm.com

ABSTRACT

With the development of Internet and social media, large-scale legal resources are available now and are very helpful for legal professionals. Although many law application solutions can provide information management and search services, it is still a big challenge to find the relevant law provisions to the complex law cases. Due to the semantic complex of various law provisions and law cases, traditional methods cannot precisely characterize the deep semantic distribution of legal cases. In this paper, we propose a neural attention model for automatically matching reference law provisions. In this proposed model, we employ word by word attention mechanism to calculate pairwise comparisons between cases and law provisions and then an output LSTM layer is used to summarize the comparisons and output the labels. The experimental results show that our model performs better than both traditional SVM classification algorithm and LSTM representation model.

CCS Concepts

•Information systems → Expert systems; •Computing methodologies → Natural language processing; *Neural networks;*

Keywords

legal text matching, neural attention model

1. INTRODUCTION

Internet and social media are changing the working styles of legal professionals. Large-scale online laws, reference law cases and other legal documents are available in the internet and social media. Such large-scale rich resources are very helpful for judges, lawyers, and other legal professionals.

Most of the law application solutions can provide information management and search services on regulation, law and reference cases. They effectively enhance legal professionals' working efficiency. However, legal research is still an expensive and time consuming process, for lawyers and judges. Lawyers and judges often spend a lots of time to prepare the law cases. For each case, they need to analyse the detail facts, collect amounts of evidences and find the relevant law clauses, reference cases and other legal citations and supports. Currently, it is still a big challenge how to quickly find the relevant law clauses to the complex law cases. Each complex law case may be related to several law provisions and the reference law provisions may be quite different because of a little differences of case details. There are also latent relevant law provisions. For example, if the defendant in a case turns himself in, the case may be relevant to probation provisions and probation period provisions. Due to the semantic complex of various law clauses, law cases, traditional rule-based or keyword-based methods can not effectively characterize the deep semantic distribution of the legal documents. Hence, we attempt to employ deep learning technology to capture both explicit and hidden semantic association among various high-level law clauses, hypothesis and the details of law cases.

We propose a deep semantic match algorithm for automatically matching reference law provisions. We define the task as a classification task to label matching or not, given a case brief and a law provision. A neural attention model is proposed to learn automatically from training data. In the proposed model, we first employ word by word attention on top of two LSTM encoders which encoding case briefs and reference law provisions respectively and calculate attention weights and comparisons of each case word's representation and law provision' representation vector after the encoding process. Then an output LSTM layer is used to summarize the compare results and output the labels. The experimental results show that our model performs better than both traditional SVM classification algorithm and LSTM representation model.

The novelties and main contributions of this paper are

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

summarized as follows:

- To the best of our knowledge, this is the first work that explore matching reference law provision task.
- We present a new neural attention model to label reference law provision.
- We report empirical results on a real legal judgement datasets and show the effective performance for this task.

The proposed algorithm is integrated into our semantic compliance advisor solution. Our compliance advisor solution focuses on semantic compliance checking on contracts, regulation and law cases, which provides semantic relatedness detection and comparison, relevant reference case finding and compliance checking for legal professional. It can effective help legal professionals to prepare the law cases and reduce the compliance risk in their daily work.

2. RELATED WORK

Recurrent neural networks(RNNs) have been used to improve language model [6] and sentence embedding[7]. RNN inputs text sequentially by taking a single token at each time step and producing a corresponding hidden state. The hidden state is then passed along through the next time step to provide historical sequence information.

Although a great success in a variety of tasks[2, 5], RNNs have limitations. It is difficult to train RNNs with the standard affine hidden units on long input sequences because the gradients tend to either vanish or explode. Because RNNs has to compress all information captured in the past time steps into the current fixed length vector, it is not good at memorizing long or distance sequences [10] Researches have proposed to overcome the limitations. The gradient exploding problems is addressed by a gradient clipping method [8] . To deal with the vanishing gradients, gated and interal short-term memory variants of an RNN unit have shown to be effective. [4, 3]. In machine translation task, in order to improve the result in translating longer sentences, Research [1] propose to use attention in RNN models, which allows RNNs to selectively focus on the most task-relevant parts of input sequence, assign importance weights to those parts and join them into a single representation.

Our method is motivated by the central role played by the neural attention mechanism in machine translation [1]. We use a neural attention model to finding reference law provision according to case briefs.

3. APPROACH

3.1 Problem Definition

As mentioned above, our task is to find reference law provision of the facts of cases. In this paper, we define the problem as a binary classification task. Given a case and a law provision, the model will give a label of matching or not.

Then we can consider the learning-based classification tasks. The training set consists of N examples $\{A^i, B^i, Y^i\}_{i=1}^N$, where the input case A^i is a sequence of word tokens $a_1^i, a_2^i, ...a_{l_a}^i$ and the input law provision B^i is a sequence of word tokens $b_1^i, b_2^i, ...b_{l_b}^i$ and the output $Y^i = (y_1^i, y_2^i)$ is an indicator vector encoding the label and the number of output classes is 2 in this paper. We assume that each word token a^i, b_i is represented by its word embedding vector $\mathbf{a}^i, \mathbf{b}^i \in R^d$ of dimension d. At test time, we receive a case A and a law provision B and the goal is to predict the correct label Y

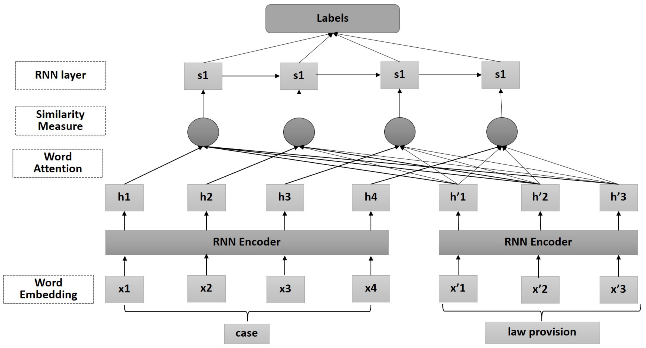

Figure 1: The architecture of Neural Attention Model

3.2 A neural attention model for matching reference law provision

The core model consists of the following three components(see Figure 1), which are trained jointly: word by word attention, compare measurement and RNN output layers.

3.2.1 *Word by word attention.*

Attentive neural networks have recently demonstrated success in a wide range of tasks such as machine translation[1], image captioning[11] and sentence inference[9]. The idea is to allow the model to attend over past output vectors. In this paper, for matching reference law provisions, we use word by word attention mechanism to soft align words in the law provisions and cases. We use two LSTMs to encode cases and law provisions respectively. Then each output state of the first LSTM attends the second LSTM's output vector and attention weights α_t and comparisons are calculated over all output vectors of law provisions for each output state h_t in the case brief.

This can be modeled as follows:

$$h_k^a = (\mathrm{Comp}(\Sigma_{j=1}^m \alpha_{kj} h'_j, h_k), \Sigma_{j=1}^m \alpha_{kj} \mathrm{Comp}(h'_j, h_k) \quad (1)$$

$$\alpha_{kj} = \frac{exp(e_{ij})}{\Sigma'_j exp(e_{kj'})} \quad (2)$$

$$e_{kj} = W^e \cdot \tanh(W^s h'_j + W^t h_k + W^c \mathrm{Comp}(h'_j, h_k)) \quad (3)$$

where h_k^a is the compare result of output state h_k of case brief and the output vectors of law provision h', all matrices W contain weights to be learned and the function Comp is the compare function we will explain next. Note that the compare result h_k^a is an concatenation of two parts: 1) compare result of h_k and an aggregate of vector h' with the attention weights; 2) an aggregate of compare result of h_k and each state h'_j. We also assume the soft align weight α_{kj} is relevant to the comparison of h_k and h'_j.

3.2.2 *Compare measurement.*

We define the compare function for comparing two states as follows:

$$\mathrm{Comp(x, y)} = (\cos(x, y), \mathrm{L2Euclid}(x, y), |x - y|, x * y) \quad (4)$$

where cosine distance(cos) and element-wise multiplication measure the distance of two vectors according to the angle between them, L2 Euclidean distance (L2Euclid) and element-wise absolute distance difference measure magnitude differences. In another perspective, cosine distance and L2 Euclidean distance measure the sum distance and element-wise multiplication and absolute distance measure the element-wise distance in each dimensions of the vectors. In this paper, L2 is set as 0.01.

3.2.3 RNN output layers.

After word by word attention, each output state of cases has a compare result with the law provision and then another LSTM is used to sequentially summarize the compare results. We take the last state of the LSTM as compare result representation and on top of the LSTM layer, we use a linear layer and a log-softmax layer as the final output layer, which outputs the matching or not label.

3.3 Network learning

To train our models, we use stochastic gradient descent (SGD) to minimize the negative log likelihood loss function and the backpropagation algorithm to compute the gradients. For the output layer, we employ dropout with a constraint on l_2-norm of the weight vectors. The dropout rate is 0.2 and the l_2 constraints is 3. Training is done through stochastic gradient descent over mini-batches with the size of 50 and Adadelta update rule[12] and the number of epochs is set to 200. The encoding LSTM layer we use in this paper are bidirectional LSTM and the numbers of cell are all set as 20. The hidden state in the output linear layer is set as 50. In our experiment, we use initialized randomly word vectors for each word and learn them as parameters during training. The dimension of word vectors is set to 50.

4. EXPERIMENT

4.1 Dataset

As there is no benchmark or published dataset for reference law provision matching, we conduct experiments on a real-world dataset collected from a set of Chinese credit card fraud judgements. We extracted the case briefs and reference law provisions from each judgement as positive samples. Table 1 shows an example case briefs and its two reference laws. We also randomly construct negative samples with case briefs and non-reference law provisions about credit card fraud.

We use 10000 case brief and law provision pairs as training set, 2000 pairs as validation set and 4000 pairs as test set. In each set, the positive-negative ratio is about 1:1. Then we do word segmentation on the datasets and use accuracy as our evaluation metric.

4.2 Compared models

To compare with the proposed model, we also evaluated the following methods on the constructed dataset.

- **SVM**. We use TFIDF of words to represent case briefs and law provisions, the output of compare function we defined in this paper as features and SVM algorithm to do classification.
- **LSTM**. We use two LSTMs to represent case briefs and law provisions respectively, a compare layer of the function we defined in this paper with the last state of the two LSTM as input to give compare results and last a linear layer and a log-softmax layer to give the label.

4.3 Experiment results and analysis

Model	accuracy
SVM	0.806
LSTM	0.891
Our model	0.911

Table 2: Experimental results.

Model	SVM	LSTM	Our model
case brief with provision 1	not-matching	matching	matching
case brief with provision 2	not-matching	not-matching	matching

Table 3: Labels of the case brief and reference law provisions in Table 1 from SVM, LSTM and out neural attention model.

Tabel 2 shows the comparison of the proposed neural attention model with compared method on the constructed evaluation dataset. We have the following observations: 1) First of all, we can see models with RNN encoder (LSTM and our neural attention models) perform better than SVM. This shows the effectiveness of learning represent by RNN encoder. 2) Our proposed neural attention model performs better than LSTM model. This shows the effectiveness of learning word soft-alignment by attention models.

Furthermore, Table 3 shows labels of the case brief and reference law provisions in Table 1 from SVM, LSTM and our neural attention model. We can find the SVM failed to match provision 1 and provision 2, LSTM matched provision 1 but failed on provision 2 but our model can match both provision 1 and provision 2. This may because LSTM can learn important features from corpus automatically. But the global-level representation cannot match provisions such as provision 2 with probation period which is latent relevant to the case brief because of the defendant's confessing. But in our proposed attention model, each word is compared to the law provision through word by word attention mechanism and the provision 2 may be matched because word *confessing*, *probation* in the case brief are related. This suggests in matching reference law provision task, pairwise comparison by attention mechanism are relatively more import than global-level representations.

5. CONCLUSION

How to quickly find relevant law provisions to the complex law cases is a big challenge due to the semantic complex of various law provisions and case details. In this paper, we propose to use neural attention model for automatically matching reference law provisions. In the proposed model. we first use two LSTM encoders to represent case briefs and law provisions respectively and employ word by word attention mechanism. At last, an output LSTM layer is used to

Case brief	Reference law provision 1	Reference law provision 2
本院认为，被告人吴某以非法占有为目的，恶意透支信用卡，数额较大，扰乱了正常的金融秩序，已构成信用卡诈骗罪。公诉机关指控被告人犯信用卡诈骗罪的事实清楚，证据确实充分，其指控罪名成立。鉴于被告人有自首情节，所透支的银行本金及利息等已全部归还，诉讼中预缴了罚金，对其适用缓刑符合相关法律规定(The Court finds that, on the purpose of illegal possession, defendant Wu maliciously overdraft credit card in relatively large quantities, disrupted the financial order and thus constituted the crime of credit card fraud. Prosecutor accused of criminal facts are clear and sufficient evidence, convicted. In view of his confessing, repayment of principal and interest and pre-paying fines in the proceedings, he is applied to probation.)	第一百九十六条有下列情形之一，进行信用卡诈骗活动，数额较大的，处五年以下有期徒刑或者拘役，并处二万元以上二十万元以下罚金；数额巨大或者有其他严重情节的，处五年以上十年以下有期徒刑，并处五万元以上五十万元以下罚金；...(Article 196 Whoever commits fraud by means of a credit card in any of the following ways shall, if the amount involved is relatively large, be sentenced to fixed-term imprisonment of not more than five years or criminal detention and shall also be fined not less than 20,000 yuan but not more than 200,000 yuan; if the amount involved is huge, or if there are other serious circumstances, he shall be sentenced to fixed-term imprisonment of not less than five years but not more than 10 years and shall also be fined not less than 50,000 yuan but not more than 500,000 yuan;...)	第七十三条拘役的缓刑考验期限为原判刑期以上一年以下，但是不能少于二个月。有期徒刑的缓刑考验期限为原判刑期以上五年以下，但是不能少于一年。 缓刑考验期限，从判决确定之日起计算。(Article 73 The probation period for suspension of criminal detention shall be not less than the term originally decided but not more than one year, however, it may not be less than two months. The probation period for suspension of fixed-term imprisonment shall be not less than the term originally decided but not more than five years, however, it may not be less than one year. The probation period for suspension of sentence shall be counted from the date the judgment is made final.)

Table 1: .An example of case briefs and reference law.

summarize the compare result calculated from the attention and output the label. We evaluated our model in real dataset from Chinese legal judgements. The experiment result show that our model performs better than SVM classification algorithm and LSTM model.

In this paper, we try to match reference law provisions with case briefs which are summarized by human. Future work will focus on matching relevant law provisions according to original case facts.

6. REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[3] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, October 2014.

[4] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003.

[5] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[6] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048, 2010.

[7] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward. Deep sentence embedding using the long short term memory network: Analysis and application to information retrieval. *In Proceedings of the 31th International Conference on Machine learning*, 2015.

[8] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318, 2013.

[9] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

[10] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[11] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2048–2057, 2015.

[12] M. D. Zeiler. Adadelta: an adaptive learning rate method. *CoRR*, 2012.