Argument Component Classification with Context-LSTM

Chao-Ming Wang^{1,a,*}, Xu-Dong Chen^{1,b}

¹School of Software, Beijing Jiaotong University, China

^aoujago@gmail.com, ^bchenxd@bjtu.edu.cn

*Corresponding author

Keywords: Argumentation Mining, Argument Component Classification, LSTM.

Abstract. Argumentation Mining (AM) is an emerging field of natural language understanding. It extends sentiment analysis, topic modeling and other existing text mining methods, aimed at tap the latent logical relationship between sentences. At present, Argument Component Classification (ACC) is a challenging subtask in AM. Existing popular SVM based models heavily rely on artificially constructed and domain dependent features, and they cannot nicely model the context relations among sentences. In this paper, we propose an ACC method based on Context-LSTM, which do not require any artificial features. Moreover, Context-LSTM can model the contextual information of the current sentence very well. We conduct experiments on two well-annotated corpora and get state-of-art results both.

基于Context-LSTM的论辩元素分类

王超名^{1,a,*}, 陈旭东^{1,b}

¹北京交通大学软件学院, 北京, 中国

^aoujago@gmail.com, ^bchenxd@bjtu.edu.cn

*通讯作者

关键词: 论辩挖掘; 论辩元素分类; LSTM

中文摘要. 论辩挖掘是近年来自然语言理解领域的一个崭新领域, 它扩展了现有情感分析、主题模型等文本挖掘方法, 旨在挖掘句子间潜在的逻辑关系。论辩元素分类是目前论辩挖掘具有挑战的一个子任务。现有流行的基于SVM的模型严重依赖于人工构造的、具有领域依赖性的特征, 且未能很好地建模句子间的上下文语境依赖关系。本文提出了基于Context-LSTM的论辩元素分类方法, 不需要繁重的人工特征设计, 而且能很好地建模当前句子的语境信息。本文在两个标注良好的数据集上进行评测, 并都取得了state-of-art的结果。

1. 引言

随着大数据时代的到来, 网络成为了人们获取信息和发表观点的重要平台。互联网上因此产生了大量非结构化或半结构化的信息, 于是迫切地需要计算机快速、有效地完成文本信息的结构化的抽取和分类。论辩挖掘(Argumentation Mining, 也称为Argument Mining)[2,3,7,9]因为专注于语音、文本等自然语言中的论辩逻辑的结构化抽取与挖掘, 近年来引起了学术界和工业界的广泛关注。

1.1 论辩

论辩(Argumentation)是用户评论中广泛存在的一种模式[1]。给出理由来支持或反驳一个观点，以说服已知或未知的受众的行为或过程，称为论辩[2]。心理学、语言学以及逻辑学上的研究[1]表明，人们在发表评论时常常使用论辩的方法来阐明自己的观点。比如“餐馆位置很好，就在西直门地铁站旁边”这句评论就蕴含着论辩形式：首先阐明自己的观点“餐馆位置很好”，然后给出理由“就在西直门地铁站旁边”来支持观点。

一个完整的论辩结构被称为一个论辩体(Argument)，由一个论点(Claim，也称为观点)，若干个论据(Premise，也称为理由、前提或证据)以及论据与论点之间的论证关系(Inference Relation)构成。而有时所有的论点也会成为论据，此时称为子论点或分论点，去支持一个中心论点(Major claim，也称为总结论、中心观点)。中心论点、分论点、论据统称为论辩元素(Argument Component)。另外，论辩体之间也可以存在一定的逻辑关系，比如支持(Support)关系、攻击(Attack)关系等。在如图1(a)所示的评论中，分论点“人非常好”和分论点“很好吃”支持着中心论点“整体不错”，而分论点“包装垃圾”则与中心论点是攻击关系。其整体论辩结构如图1(b)所示。

图 1 论辩示例

1.2 论辩挖掘

论辩挖掘正是从诸如上述半结构化或非结构化文本中自动抽取结构化论辩体，以及论辩体之间的关系与结构的一门崭新的研究课题。它扩展了现有的文本挖掘方法，比如情感分析、主题模型等，揭示了句子间潜在的逻辑关系。图2比较了几种现有文本挖掘方法和论辩挖掘方法在分析结果上的不同。

语句	主题模型	情感分析	论辩挖掘
餐馆位置很好，就在西直门地铁站旁边。	主题： 交通、餐馆	餐馆位置很好[正向情感]，就在西直门地铁站旁边[中性情感]。	餐馆位置很好[Claim]，就在西直门地铁站旁边[Premise]。 support

图 2 不同的文本挖掘方法对用户评论的分析结果

论辩挖掘是自然语言理解领域的重要研究分支，涉及统计学、心理学、逻辑学、语言学、计算论辩学(Computational Argumentation)等多学科的知识与理论，同时涉及信息检索(Information Retrieve)、知识表示(Knowledge Representation)、篇章分析(Discourse Analysis)等多领域的技术与方法[3]，具有很大的研究和应用价值。比如在舆情监控应用中，论辩挖掘可以辅助决策者在观察舆情变化趋势的同时洞察其背后的原因，从而更加准确地了解网络用户的意见和行为；在个性化推荐应用中，论辩挖掘可以帮助推荐系统在给出推荐结果的同时解释推荐原因，实现“可解释的推荐”。2014年IBM甚至花费几百万美元开发了一个基于认知计算的论辩挖掘商业项目，The Debater Project。可见论辩挖掘蕴含着巨大的商业应用前景。

1.3 论辩挖掘的任务

经过近几年的研究，论辩挖掘问题已将发展为一个多任务的递进式问题。目前已知的论辩挖掘系统大多是多阶段的流水线系统。一个阶段面对一个子任务，解决一个子问题。而系统每一阶段的输入又是前一阶段的任务输出。论辩挖掘的主要任务如下：

任务一：论辩边界的划分(Argument Boundaries Detection, ABD)，即划分出正确的句子边界，使边界内的句子只包含一个论辩元素，用于下一阶段论辩元素的分类。通常情况下，一个句子只包含一个论辩元素，但有时，一个包含论辩元素的句子可能并不都是具有论辩逻辑的，如图3(a)；或者，一个论辩元素可能由多个句子组成，如图3(b)；甚至，一个句子内包含多个论辩元素，如图3(c)。因此，正确的边界划分是论辩元素分类的前提。

还行吧[Claim]，但是感觉还是没麦当劳什么的好吃[Premise]。	(a) Premise 所在句并不都是论辩元素
发票写错了，联系客服，竟然联系不到店家，也退不了单。 [Premise] 这服务真是太差劲了。[Claim]	(b) Premise 由多个句子组成
正是因为太油腻了[Premise] 所以我说这菜不好吃[Claim]。	(c) 该句内包含多种论辩元素

图 3 三种不同情况下的论辩元素的抽取

任务二：论辩元素的分类(Argument Components Classification, ACC)，即识别出每个句子的论辩类别。常见的论辩元素类别包括：Major claim、Claim、Premise。但目前论辩挖掘的语料标注标准上还不是很统一，不同的语料标注出的论辩元素可能不一致。比如一些评论语料内，除了Major claim、Claim、Premise，还可能出现推荐(Recommendation)、背景(Background)等元素。在论辩元素分类下还具有更小的子任务，包括观点检测(Claim Detection)[4, 5]和前提/证据检测(Evidence Detection)[6]等。

任务三：论辩关系的识别(Argument Relationship Detection, ARD)，即从抽取出的论辩元素中识别出相互之间的支持或攻击关系，从而将抽取到的论点与论据配对，构造出论辩体。

1.4 本文工作

目前论辩挖掘亟需解决的挑战性任务是论辩元素的分类问题(任务二)[7]，本文主要面向这一工作。目前大多数是使用监督学习方法，主要工作集中在寻找有效的特征(feature)和合适的分类器(classifier)上。最常用的特征包括结构特征(诸如位置、标点)、词汇特征(n-grams、动词、副词、情态动词等)、句法特征(依存句法树、词性等)、语境特征(标点、词语、分句等的数量)等[8]，甚至包括特定文本的领域知识(比如Debater需要领域专家提供尽可能多的相关领域的知识)[5, 6]。常用的分类器包括Naive Bayes[1, 9]、SVM[8,9,10]、最大熵[10]等。但是，目前这些方法存在如下几个问题：

1. 严重依赖于人工构造的特征。人工构造特征工程量大，不仅费时费力，而且特征的构造具有领域依赖性(Domain-Dependent)，不同的数据构造的特征不尽相同[4]。
2. 较难考虑句内的长距离依赖关系和句间的上下文信息。基于“特征工程+分类器”的方法可以通过一个词语附近固定窗口的词语进行n-grams建模，却很难考虑到长距离依赖关系。同时，同一句子在不同的上下文信息下，可能具有不同的论辩元素类型。

为了克服以上几个问题，我们提出了基于Context-LSTM的论辩元素分类方法，其主要优势在于：(1)、不需要繁重的人工特征设计，能进行端对端地学习；(2)、基于LSTM的ACC模型能利用LSTM的内部记忆单元，有效地建模句子内部的序列依赖关系；(3)、基于LSTM改进的Context-LSTM的ACC模型能有效地建模上下文信息，并捕捉到当前句子的语境。本文将提出的模型在两个标注良好的数据集上进行评测，并都取得了state-of-art的结果。

2. 相关工作

最早的论辩挖掘研究开始于法律文本、在线评论和辩论等特定领域[9,11,12,13]。近几年因为机器学习和自然语言处理等技术的发展以及标注语料的增多，论辩挖掘开始逐渐繁荣。主流的论辩元素抽取方法主要利用传统自然语言处理中监督性学习的方法，在每个句子只包含一个论辩元素的前提下，利用各种常见的特征集合，诸如词汇、n-grams、bag-of-words、依存句法[8,9]，以及一些高度复杂的特征，诸如主客观情感分析得分、领域知识或辞典信息[5]，来对每个句子进行分类。

另外也有极少数的一些非监督学习方法来自动抽取特征。Lawrence et al.[14]在同一段落内相同的论辩元素应该属于相同主题的假设下，运用LDA主题模型抽取每个段落的主题，然后利用主题信息作为特征进行分类。Nguyen和Litman[15]利用LDA抽取论辩性词语(诸如“Think”、“Reason”)和领域性词语(诸如“Education”、“Art”)来作为特征进行分类。

3. 论辩元素分类

本节首先简要介绍LSTM，然后介绍基于LSTM的ACC模型，最后介绍加入上下文(context)信息的Context-LSTM的ACC模型。

3.1 LSTM

长短时记忆网络(Long Short-Term Memory, LSTM)[16]的提出，是为了解决传统RNN模型在BPTT训练的时候会遭遇梯度爆炸或梯度消失的问题[17]。LSTM通过引入遗忘门(Forget gate)、输入门(Input Gate)和输出门(Output Gate)，在一定程度上克服了这个问题。LSTM可形式化表示为：

$$i_t = \delta(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \delta(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \delta(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * g_t \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

其中， δ 表示sigmoid激活函数， W 、 U 为输入层到隐含层、隐含层到隐含层的系数矩阵， b 为偏置项。 i_t 、 f_t 、 o_t 分别为输入门、遗忘门、输出门。而 c_t 是LSTM的关键，称为内部记忆状态(Internal Memory State)，它贯穿LSTM的始终。 h_t 表示隐含层的输出。在LSTM中，遗忘门 f_t 决定了有多少上一个时刻的记忆可以保留，输入门 i_t 决定了有多少当前时刻的状态可以被记忆，输出门 o_t 决定了有多少当前时刻的记忆可以被输出。而 g_t 则是综合考虑当前输入和上一时刻输出后的临时记忆状态，为之后的内部记忆状态 c_t 的计算提供支持。

3.2 基于LSTM的ACC模型

若一篇文章由多个句子组成， $docX = \{sen_1, \dots, sen_m, \dots, sen_M\}$ ，每个句子对应一个标签， $docY = \{y'_1, \dots, y'_m, \dots, y'_M\}$ 。同时每个句子包含多个词， $sen_m = \{x_1, \dots, x_t, \dots, x_T\}$ ，其中 x_t 为第 t 个词对应的词向量，则基于LSTM的ACC框架(如图4)可形式化表示为：

$$o_m = LSTM([x_1, \dots, x_t, \dots, x_T]) \quad (7)$$

$$y_m = softmax(W_y * o_m + b_y) \quad (8)$$

图 4 基于LSTM的ACC模型

图 5 LSTM Encoder

图 6 上下文窗口为1的Context-LSTM模型

其中， $LSTM$ 表示基于LSTM的句子编码器； o_m 表示句子 sen_m 被编码后的表征向量； W_y 是 $softmax$ 层的系数矩阵， b_y 表示偏置系数。

基于LSTM的句子编码器让句子内每个词的词向量顺序通过，经过LSTM的中心单元Cell的非线性转换后(Cell的计算见公式(1)-(6))，得到词语级的表征 h_t ，词语级的表征通过累加再平均，得到句子级的表征 o_m ，如图5。形式化表示 $LSTM Encoder$ 如下：

$$h_t = Cell(x_t) \quad (9)$$

$$o_m = AVG(\sum_{t=0}^T h_t) \quad (10)$$

在训练时，我们采用交叉代价熵（Cross-Entropy）作为目标函数，并且采用 L1 和 L2 正则项，

$$C = -\sum_{w_m} [y'_m \ln y_m + (1 - y'_m) \ln(1 - y_m)] + \partial 1 \sum_{s \in W, b} \|s\| + \partial 2 \sum_{s \in W, b} s^2 \quad (11)$$

其中 $\partial 1$ 和 $\partial 2$ 分别为 L1 和 L2 正则项的系数。 W 、 b 分别表示框架内所有的系数矩阵和偏置系数。

3.3 基于Context-LSTM的ACC模型

上下文语境在论辩挖掘中具有重要地位。比如，在不同的语境下，即便是同一个句子也可能具有不同的论辩分类。因此，正确地考虑上下文信息是提高论辩元素分类精度的关键。本文在 $LSTM Encoder$ 的基础上，来考虑论辩元素分类的上下文信息。在 $softmax$ 层进行当前句子的分类时，我们除了利用当前句子的表征向量 o_m ，还加入了该句的上下文窗口大小为 c 的句子表征向量 $[o_{m-c}, \dots, o_{m-1}, o_{m+1}, \dots, o_{m+c}]$ 。其目的是利用上下文句子的表征向量中蕴含的语义信息来辅助当前句子的分类决策。可形式化表示Context-LSTM的 $softmax$ 层为：

$$y_m = softmax(W_y * o_m + W_{yc} * [o_{m-c}, \dots, o_{m-1}, o_{m+1}, \dots, o_{m+c}] + b_y) \quad (12)$$

其中， o_m 为当前句子 sen_m 的句子级表征， o_{m-c} 表示前文第 c 个句子的表征， o_{m+c} 表示后文第 c 个句子的表征， W_{yc} 则表示当前句子context信息的矩阵系数。图6所示为上下文窗口为1的Context-LSTM模型。

4. 实验

4.1 数据和环境

为了验证模型的有效性，我们在两个数据集上进行了评测：Persuasive Essays (PE)数据集[18]和Hotel数据集[19]。PE数据集包含402篇说服性文章，总共7,116条句子和147,271个词语，其中包含了三种论辩元素：Major claim、Claim和Premise。而Hotel数据集有五种论辩元素：Major claim、Claim、Premise、Background和Recommendation。该数据集一共包含1,575条句子和14,756个单词。这两个数据集是目前已知的公开数据集中标注质量最好的数据[19]。

本文基于Theano深度学习框架实现上述模型。我们使用Google News Word Embedding进行词向量的初始化。另外，*LSTM Encoder*使用两层，其隐含层大小为300，BPTT步长设置为6。本文也采用dropout进行正则化，并使用Adam优化方法进行模型的训练。

4.2 实验设计和结果

在Hotel数据集中，我们采用十折交叉验证法进行评测；而PE数据集因为提供了其中的80篇作为默认的标准测试集，因此采用留出法进行评测。在评价指标上，本文与文献[18,19]保持一致，包括Macro F_1 以及各数据集上单一类别的 F_1 值，即：MC (Major claim类别的 F_1 值)、Cl (Claim类别的 F_1 值)、Pr (Premise类别的 F_1 值)、Bg (Background类别的 F_1 值)、Re (Recommendation类别的 F_1 值)、以及Macro (整体类别的 F_1 值)。

在实验初期，我们尝试与SVM模型进行比较。SVM使用的特征与文献[8,18,19]保持一致，而Context-LSTM采用上下文窗口为1。从表1可以看出，SVM在Major claim元素的分类上相比LSTM有一定的优势。其原因可能是因为Major claim的人工特征较明显且有效。而在加入上下文信息之后，Context-LSTM相比LSTM在Major claim上的表现有明显好转，在Hotel数据集上甚至显著超越SVM模型。从表1也可以看出，Context-LSTM能有效地建模到上下文信息，几乎在各个指标上都比LSTM表现要好。同时，其整体性能也要优于基于特征的SVM模型。

表 1 各种模型在两个数据集上的实验结果

模型	PE 数据集				Hotel 数据集					
	Macro	MC	Cl	Pr	Macro	MC	Cl	Pr	Bg	Re
SVM[18, 19]	0.794	0.891	0.611	0.879	0.578	0.831	0.635	0.413	0.654	0.633
LSTM	0.793	0.825	0.645	0.891	0.532	0.569	0.685	0.459	0.614	0.660
Context-LSTM	0.801	0.829	0.679	0.895	0.580	0.925	0.704	0.495	0.598	0.654

本文也实验了context窗口的大小对于模型性能的影响。我们选取了上下文窗口为1、2、3进行实验。在保持其他条件与上述实验一致的情况下，我们得到了表2的实验结果。从表2中，我们可以发现，Context-LSTM的性能与上下文信息的多少呈正相关关系。本文也由此推测，最理想的上下文信息量应该为全文信息，即在每个句子进行分类决策时都考虑到篇章整体的语义信息。另外，在本实验中，Context-LSTM模型在两个数据集上都取得state-of-art的结果，在PE数据集中Macro F_1 达到81.4%，在Hotel数据集中Macro F_1 达到61.7%。

表 2 不同上下文窗口大小下的Context-LSTM的实验结果

上下文窗口	PE 数据集				Hotel 数据集					
	Macro	MC	Cl	Pr	Macro	MC	Cl	Pr	Bg	Re
1	0.801	0.829	0.679	0.895	0.580	0.925	0.704	0.495	0.598	0.654
2	0.808	0.843	0.680	0.901	0.596	0.930	0.679	0.488	0.598	0.735
3	0.814	0.844	0.687	0.912	0.617	0.925	0.669	0.544	0.661	0.768

5. 结束语

本文对论辩挖掘进行了介绍。并针对论辩挖掘中目前具有挑战的论辩元素分类任务提出了Context-LSTM模型。相比于传统SVM模型，Context-LSTM模型无需人为提取文本中蕴含的特征，并能充分利用句子间的上下文语境信息。在两个数据集上的实验结果表明了上下文信息的重要性，同时Context-LSTM模型也在两个数据集中取得了state-of-art的结果。

致谢

本文为北京交通大学大学生科研训练资助项目(201710004066)。另外，非常感谢赵晓朝在论文写作过程中给予的巨大帮助。

References

- [1] Mercier, Hugo, and Dan Sperber. "Why do humans reason? Arguments for an argumentative theory." *Behavioral and brain sciences* 34.02 (2011): 57-74.
- [2] Moens, Marie-Francine. "Argumentation Mining: Where are we now, where do we want to be and how do we get there?." *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*. ACM, 2013.
- [3] Lippi, Marco, and Paolo Torroni. "Argument Mining: A Machine Learning Perspective." *International Workshop on Theorie and Applications of Formal Argumentation*. Springer International Publishing, 2015.
- [4] Lippi, Marco, and Paolo Torroni. "Context-independent claim detection for argument mining." *Proceedings of the Twenty-Fourth International Conference on Artificial Intelligence*. 2015.
- [5] Levy, Ran, et al. "Context dependent claim detection." *Citeseer*, 2014.
- [6] Rinott, Ruty, et al. "Show Me Your Evidence—an Automatic Method for Context Dependent Evidence Detection." *Proceedings of the 2015 Conference on Empirical Methods in NLP (EMNLP)*, Lisbon, Portugal. 2015.
- [7] Lippi, Marco, and Paolo Torroni. "Argumentation mining: State of the art and emerging trends." *ACM Transactions on Internet Technology (TOIT)* 16.2 (2016): 10.
- [8] Stab, Christian, and Iryna Gurevych. "Identifying Argumentative Discourse Structures in Persuasive Essays." *EMNLP*. 2014.
- [9] Mochales, Raquel, and Marie-Francine Moens. "Argumentation mining." *Artificial Intelligence and Law* 19.1 (2011): 1-22.
- [10] Park, Joonsuk, and Claire Cardie. "Identifying appropriate support for propositions in online user comments." *Proceedings of the First Workshop on Argumentation Mining*. 2014.
- [11] Cabrio, Elena, and Serena Villata. "Natural Language Arguments: A Combined Approach." *ECAI*. Vol. 242. 2012.
- [12] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2010): 1345-1359.
- [13] Moens, Marie-Francine, et al. "Automatic detection of arguments in legal texts." *Proceedings of the 11th international conference on Artificial intelligence and law*. ACM, 2007.
- [14] Lawrence, John, et al. "Mining arguments from 19th century philosophical texts using topic based modelling." *Proceedings of the First Workshop on Argumentation Mining*. 2014.
- [15] Nguyen, Huy V., and Diane J. Litman. "Extracting argument and domain words for identifying argument components in texts." *Proceedings of the 2nd Workshop on Argumentation Mining*. 2015.
- [16] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [17] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." *IEEE transactions on neural networks* 5.2 (1994): 157-166.
- [18] Stab, Christian, and Iryna Gurevych. "Parsing argumentation structures in persuasive essays." *Computational Linguistics* (2017).
- [19] Gao, Y., Wang, H., Zhang, C., & Wang, W. (2017). Reinforcement Learning Based Argument Component Detection. *arXiv preprint arXiv:1702.06239*.