Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Structured prediction models for argumentative claim parsing from text

Filip Boltužić & Jan Šnajder

To cite this article: Filip Boltužić & Jan Šnajder (2020) Structured prediction models for argumentative claim parsing from text, Automatika, 61:3, 361-370, DOI: 10.1080/00051144.2020.1761101

To link to this article: https://doi.org/10.1080/00051144.2020.1761101

Published online: 12 May 2020.

AUTOMATIKA
2020, VOL. 61, NO. 3, 361–370
https://doi.org/10.1080/00051144.2020.1761101

Taylor & Francis
Taylor & Francis Group

REGULAR PAPER

OPEN ACCESS Check for updates

Structured prediction models for argumentative claim parsing from text

Filip Boltužić and Jan Šnajder

Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

ABSTRACT
The internet abounds with opinions expressed in text. While a number of natural language processing techniques have been proposed for opinion analysis from text, most offer only a shallow analysis without providing any insights into *reasons* supporting the opinions. In online discussions, however, opinions are typically expressed as arguments, consisting of a set of claims endowed with internal semantic structure amenable to deeper analysis. In this article, we introduce the task of *argumentative claim parsing (ACP)*, which aims at extracting semantic structures of claims from argumentative text. The task is split into two subtasks: *claim segmentation* and *claim structuring*. We present a new dataset on two discussion topics with claims manually annotated for both subtasks. Inspired by structured prediction approaches, we propose a number of supervised machine learning models for the ACP task, including deep learning, chain classifier, and joint learning models. Our experiments reveal that claim segmentation is a relatively feasible task, with the best-performing model achieving up to 0.37 and 0.79 exact and lenient macro-averaged F1-score, respectively. Claim structuring, however, proved to be a more challenging task, with the best-performing models achieving at most 0.08 macro-averaged F1-score.

ARTICLE HISTORY
Received 19 February 2020
Accepted 23 April 2020

KEYWORDS
Opinion mining; argumentation mining; natural language processing; machine learning; deep learning; structured prediction

1. Introduction

Public opinion drives many decisions in politics, governance, business, marketing, and many other areas. The internet has become the most dominant source of opinion, especially since the birth of social media [1, 2]. Massive amount of opinions in text have given rise to natural language processing (NLP) techniques such as opinion mining and sentiment analysis [3], which aim to automatically classify opinions in text as being either positive or negative towards an attitudinal target (e.g. a person or a product). A related approach is that of *stance detection* [4], in which the opinion is framed as a bipolar stance that is either in favour (PRO) or against (CON) a particular (often controversial) topic [5]. While both techniques have a number of practical applications, the analysis they offer is shallow in that they typically do not provide any insights into the *reasons* underlying an opinion.

The question of how different reasons combine to form opinions falls within the purview of the field of *argumentation* [6]. From an argumentative point of view, each stance is typically supported by *arguments* [7], consisting of a network of linked *claims* [6], i.e. statements one wants others to accept and act upon [8]. For instance, a comment on social media on the topic of *marijuana legalization* might look as follows:

> *Smoking pot is bad for your health. Therefore, we should criminalize marijuana. Disallowing marijuana will improve public health.*

This argument may be broken down into three claims:

(1) *Smoking pot is bad for your health.*
(2) *Therefore, we should criminalize marijuana.*
(3) *Disallowing marijuana will improve public health.*

The claim "*therefore, we should criminalize marijuana*" expresses a CON stance towards *marijuana legalization*, while the two other claims serve to back up that stance. Thus, to determine the arguments (i.e. reasons) behind a stance, one needs to determine the argumentation structure of the comment. Each claim, however – being a natural language statement – does have an internal semantic structure. In fact, most argument-relevant claims express semantic relations between semantic concepts, representing a proposition about the world that the opinion holder believes or desires to be true. By uncovering this internal structure, we can analyse in more detail the beliefs and values of the opinion holders. For instance, we can investigate whether people who think that marijuana has no harmful health effects also think that it should be legalized, analyse how many people support marijuana legalization based on the argument that it could generate revenues if taxed, or determine the specific points on which two opinion holders disagree. Furthermore, by considering semantic and logic relations between claim structures, we could infer claims that are implicitly entailed from the

CONTACT Jan Šnajder jan.snajder@fer.hr Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, Zagreb10000, Croatia

opinion holder's claim, e.g. the claim *Not smoking pot improves public health* in the example above. In sum, an analysis of the semantic structure of claims allows for a more detailed and insightful of opinions expressed in text.

Aiming to address this need, in this article, we consider a novel NLP task of *argumentative claim parsing (ACP)*. We define ACP as a task of automatic extraction of semantic structures of claims from argumentative text. The task can conceptually be broken down into two subtasks: (1) *claim segmentation*, in which the text is segmented into fragments that correspond to individual claims, and (2) *claim structuring*, in which the segmented fragments are mapped to structures representing the semantic meaning of the claims. We frame the two subtasks as sequence labeling and structured prediction tasks, respectively, and experiment with several machine learning approaches. We also present a manually annotated dataset, featuring two discussion topics and designed specifically for this task, on which we train and evaluate our models. Our best-performing preliminary models achieve macro-averaged F1-score of 0.37 and 0.08 on claim segmentation and claim structuring problems, respectively. The contribution of our work are (1) the definition of ACP task, (2) an annotated dataset of claim segments and structures, and (3) preliminary structured prediction models for the ACP task.

2. Related work

The ACP task is closely related to the area of argumentation mining. We next review related work from argumentation mining.

Argumentation mining is a subfield of NLP dealing with the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language [9]. The transformation of text into argumentation structures is typically accomplished with a pipelined NLP architecture consisting of two steps: argument component extraction and argument component structuring [10], where the latter relies on some theoretical argumentation model, such as the Freeman's model [11]. While we adopt the same pipelined approach, our models work at the level of claims rather than the level of argumentation structures. Thus, from a functional perspective, the subtask of claim segmentation corresponds to argument component extraction, while claim structuring corresponds to argumentation structuring.

In general, *argument component extraction* divides the text into so-called Argumentative Discourse Units (ADU), which are minimal units of discourse [12]. One of two basic approaches are typically applied: a sentence-level approach or a token-level approach. In [13, 14], claim segmentation is done simply by assuming that each sentence is an argumentative claim. In contrast, in [15], the authors apply token-level segmentation using a conditional random field (CRF) model to identify the boundaries of argument components. A claim annotation study in [16] revealed that the majority of argument components span beyond single sentences, suggesting that token-level segmentation is more adequate than sentence-level segmentation. With this in mind, we adopt the token-level approach and allow for overlapping and discontinuous claim segments for additional flexibility.

Argument structure prediction maps extracted argument components into structures defined by an argumentation model. Most approaches adopt some variation of the Freeman's claim/premise model [12, 13]. For instance, in [17], Freeman's model is adopted and an SVM is used to predict links between claims and premises. In [18, 19], a model is defined based on binary excitatory and inhibitory relations between domain-specific concepts. We use these models as a starting point but define a different set of relations and use a hierarchical arrangement of concepts.

The tasks of argumentation mining involves the transformation of text into structured representations. These are typically solved using structured prediction, a supervised machine learning paradigm that predicts structured objects such as sequences, trees, and graphs [43] . Conditional random fields (CRF) is a very powerful class of probabilistic modelling methods used for structured prediction [20]. Whereas a classifier predicts a label for an instance independently of other instances, a CRF can account for context. CRFs, particularly linear-chain CRFs, have been widely applied in NLP. Recent approaches to structured prediction rely on deep learning models. Long short-term memory network (LSTM) [21] is a recurrent neural network architecture with feedback connections that models sequences of data. LSTM networks modelling data in both forward and backward directions (BiLSTM) are often used to solve text classification problems [22] or sequence labelling problems [23, 24]. Distributed word representations [25] are often used as input features to solve such problems [26]. A popular alternative to probabilistic and deep learning models for structured prediction is *chain classification* [27]. Since the ordering of classifiers may significantly impact performance, ensembling of chain classifiers is often employed [28].

3. Argumentative claim parsing

We define argumentative claim parsing (ACP) as the task of taking a number of sentences text as input and producing a set of *claim structures* as output. As noted in the introduction, ACP can be broken down into two subtasks: claim segmentation and claim structuring. In this section we formally define the two subtasks.

3.1. Claims and claim types

We begin with the definition of a claim. A claim is a statement that the opinion holder seeks to convince others to accept [29]. We adopt the typology of [29] and distinguish between claims of *fact*, *value*, and *policy*. A claim of fact is a potentially verifiable assertion as to the nature of things, which may be true or false, e.g. *marijuana is not a heavy drug*. A claim of value indicates a subjective preference or judgement, which can be positive or negative, e.g. *use of heavy drugs is bad*. Lastly, a claim of policy is an assertion that something should be done, often expressed with modal verbs such as "should" or "ought". The three types act as a wrapper around the propositional content of the claim, effectively modulating what is being claimed. For instance, the claims *marijuana should be legalized* and *marijuana is legalized* differ only in type (fact vs. policy), but their propositional content is the same.

3.2. Claim segmentation

A single sentence may contain several claims and, vice versa, a single claim can span several sentences. For this reason we need to segment the text in claims. More formally, let $x = (x_1, \ldots, x_N)$ represent text of length N as a vector of tokens where the subscript $i \in \{1, \ldots, N\}$ represents the position of the token in text. Then $Y = (Y_1, \ldots, Y_K)$ is a vector where each element represents a tuple of N elements for K segments. Y_k is a tuple of N values $Y_k = (Y_{k,1}, \ldots, Y_{k,N}), k \in \{1, \ldots, K\}$, where value $Y_{k,i} \in \{0, 1\}$ indicates if token x_i is a part of segment k. The **claim segmentation** problem is then defined as finding function f such that $f : \mathbf{x} \rightarrow \mathbf{Y}$, where $\mathbf{x}$ and $\mathbf{Y}$ represent the sets of texts and corresponding segments. The kth claim segment of x is then defined as $seg_k = (x_i | f(x)_{k,i} = 1, i \in \{1, \ldots, N\})$.

3.3. Claim structuring

Claim structuring maps segmented claims to structures representing the semantics of these claims. A claim structure essentially represents claim's propositional content and claim's type. The claim's propositional content is represented as a *semantic relation* between *domain concepts*. For instance, the claim segment "*marijuana smoking causes cancer*" may be mapped to a structure `causes(marijuana consumption, cancer)`.

In text, domain concepts are expressed as noun phrases or anaphoric references to noun phrases. The domain concepts may be arranged into a domain taxonomy. For instance "heavy drugs", "heroin", and "marijuana" all belong under the concept of "drug". The taxonomic relations can be used for inference over claims, e.g. to infer that an opinion holder who thinks that Marijuna should be legalized also subscribes to the belief that some forms of drugs should be legalized. The set of concepts is obviously domain-dependent and needs to be defined for each new argumentation topic.

In this work, we consider four semantic relations between domain concepts:

- `promotes` (subtyped as `causes` and `implies`),
- `suppresses` (subtyped as `does_not_cause` and `contradicts`),
- `comparison`, and
- `declares`.

The `declares` is an unary relation that indicates the existence of a domain concept. E.g. the claim "*marijuana is legal*" can be represented with `declares(legalized marijuana)`. Relations may be negated, e.g. the claim "*marijuana is not legal*" can be represented as ¬`declares(legalized marijuana)`. The `promotes` and `suppresses` relations are used to represent claims that express causal or implicative relations between concepts. The `promotes` relation is subtyped with `causes` and `implies`, whereas the `suppresses` relation is subtyped with `contradicts` and `does_not_cause`. The claim "*smoking marijuana hurts your lungs*" can then be represented as `promotes(marijuana consumption, lung damage)`. The `comparison` relation is used to formalize a comparison of two domain concepts according to a third concept as the criterion. For example, the claim "*alcohol is worse for your health than marijuana*" can be structured as: `comparison(alcohol, marijuana, negative_health_effect)`. The `declares` relation indicates an acknowledgement of existence of a domain concept. For example, the claim "*marijuana consumption is out there*" acknowledges the existence of the domain concept *marijuana consumption*.

Conceptually, claim structures are triplets consisting of an n-ary semantic relation and n domain concepts. For practical purposes, however, we decompose a claim structure representing an n-ary relation into a set of $n + 1$ triplets. The motivation for this is twofold. Firstly, this makes structured prediction with variable arity easier. Secondly, it makes the representation compatible with the well-established machine readable frameworks, such as the Resource Description Framework (RDF) [30], which could then be potentially used for inference over claim structures.

Each triplet is comprised of `claim id, relation, domain concept`, where the `claim id` uniquely identifies the claim. For `promotes` and `suppresses` relations, we introduce an auxiliary relation `has_antecedent` relation to denote the antecedent of the relation. The comparison relation is decomposed into a set of three triplets: `comparison_greater`, `comparison_less`, and `comparison_criterion`.

Finally, we define the task of claim structuring as finding a function $g : \mathbf{seg} \rightarrow \mathcal{C}$, where **seg** represents claim segments and $\mathcal{C}$ are the corresponding target structures, each consisting of a set of $n + 1$ triplets.

4. Dataset annotation

To train and evaluate structured prediction models for the ACP task, we introduce a novel dataset manually annotated with claim segments and claim structures. As a starting point, we adopt the dataset of Hasan and Ng [31], which contains user comments from two-sided online debates on a number of topics. For our work we chose the "Marijuana Legalization" and "Gay Rights" topics, from which we sampled 100 comments per topic (50 pro and 50 con), for a total of 200 comments. The 200 comments comprise of 20921 tokens (104.61 per comment) and 1173 sentences (5.87 per comment).

Annotation was carried out by three trained annotators, near-native speakers of English. The process was split into two phases: claim segmentation and claim structuring. In the first phase, annotators were asked to first segment out claims from user comments. For the second phase, claim structuring, we first compiled a list of domain concepts based on the previously annotated claim segments. This list was then used by the annotators to produce claim structures from claim segments. Due to resource constraints, the second phase was carried out only for the "Marijuana Legalization" topic. We next describe the two phases in more detail.

4.1. Claim segmentation

Annotating claim segments amounts to performing two tasks: (1) separating argumentative content from non-argumentative content and (2) combining overlapping and discontinuous text fragments into claims. From a linguistic point of view, this annotation is difficult and subjective, as there are many ways a comment can be segmented into claims. The ambiguity can be reduced by doing these two tasks jointly.

Unlike most previous token-based approaches to claim segmentation [15], we allow for both overlapping and discontinuous segments. For example, the claim "*marijuana is good for the economy and harmful for health*" is segmented to s_1 = "*marijuana is good for the economy*" and s_2 ="*marijuana is harmful for health*", where s_1 and s_2 overlap in tokens "*marijuana is*" and s_2 is discontinuous. This design choice is motivated by increased coverage, as a fair number (19.1% in our dataset) of claims are overlapping or discontinuous.

The 200 users' comments yielded 1817 claim segments (an average of 9.1 claims per comment), of which 920 for the "Marijuana Legalization" and 897 for the "Gay Rights" topic. In total, 89.74% of text is covered by argumentative segments, while 10.26% was annotated as non-argumentative.

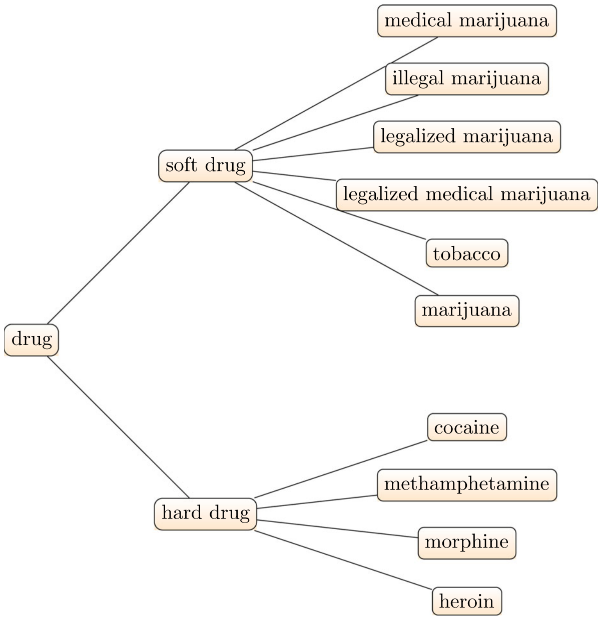

Figure 1. "Marijuana Legalization" domain concept hierarchy.

4.2. Claim structuring

In the second annotation phase, the annotators were asked to map each claim segment into a claim structure, where the structure is defined as a set of triplets (cf. Section 3.3). Since claim structures have to refer to domain concepts, we first proposed a list of these concepts based on the claim segmentation annotation. The three annotators were provided this list, but they were also instructed to propose additional domain concepts if they felt these concepts will result in claim structures that more truthfully capture the semantics of the claim. Figure 1 shows a part of the taxonomy of domain concepts rooted in the `drug` concept. We defined a total of 75 concepts, such as `marijuana addicted consumer`, `legalized marijuana`, `legalized alcohol`, and `reduced mental capability`.

Having defined a list of domain concepts, we proceeded with annotating claim structures for the "Marijuana Legalization" topic. The three annotators, each working independently, produced claim structures for 920 claims. Some of the obtained structures are shown in Table 1. The three annotators failed to produce a claim structure for 56, 53, and 60 claims, respectively, which makes about 6% of the claims. There are a number of reasons why a claim cannot be adequately represented as a claim structure (defined in the way we defined it). For one, the claim may be lacking appropriate domain concepts, such as in the claim "*tobacco odours dissipate quickly*", where *tobacco odour* was not included in the domain concept list. The rest of failures pertains to claims deemed too abstract in meaning, an example being "*nothing can bring world peace*". For 243 out of 920 claims, the same claim structure was produced by all three annotators, while for 325 claims two

Table 1. Claim structure annotation examples: the original and structured claims.

Original	Structure
Pot hurts you	*< claim_1, has_antecedent, marijuana_consumption >* *< claim_1, promotes, negative_health_effect >*
One may suffer or develop hallucinations,	*< claim_2, 0.1cm has_antecedent, marijuana_consumption >* *< claim_2, causes, mind_influential >*
Cannabis has been proven to have health benefits.	*< claim_3, has_antecedent, marijuana_consumption >* *< claim_3, suppresses, negative_health_effect >*

out of three annotators agreed. Upon manually inspecting the structures, we observe that, although some solutions were different across annotators, they all seemed equally plausible.

5. Models

5.1. Claim segmentation

We propose two ways of framing the claim segmentation problem (as defined in Section 3.2). In the first, we frame the problem as multi-label classification and train a binary relevance (`BR`) classifier that assigns a segment identifier to each token. The second approach is an alternative to inefficient `BR` labelling, where we ignore the overlapping and discontinuous segments and apply the well-established `BIO` tagging setup. `BIO` labels indicate whether the word is outside a segment (`O`), starts a segment (`B`), or continues a segment (`I`). Table 2 shows a comparison of the `BR` and `BIO` setups.

We propose three claim segmentation models. The first model, dubbed the naïve heuristics and used as the baseline, adopts a sentence splitting approach, commonly seen in argumentation mining [13, 14]. The second model is a support vector machine (SVM) with tf-idf features as input. The third model combines deep learning with structured prediction. Below we describe the two models in more detail.

5.1.1. Support vector machine

For the second approach, we use a weighted support vector machine (SVM) model [32]. To represent tokens as features, we use tf-idf and distributed word representations (fastText[1] and word2vec[2] pretrained vectors). Finally, to train the model, we use 5 × 3 nested-cross validation optimizing hyperparameters C and γ using grid search implemented in the libSVM framework [33].

5.1.2. BiLSTM-CRF

The third model combines a deep learning recurrent model (BiLSTM) and conditional random fields (CRF). The reason we opt for this model is two fold. First, BiLSTMs have previously been successfully used in argumentation mining [34] and text classification in general [35]. Second, the combination of a BiLSTM with a CRF is considered extremely effective for sequence tagging problems [36]. Our model works in two stages. In the first stage, a BiLSTM is used to encode a sequence of tokens of the comment. The BiLSTM produces pairs of hidden states and outputs. The outputs of the BiLSTM are then fed into a feed-forward linear layer, which maps the BiLSTM outputs to the label probability space. In the second stage, the output of the BiLSTM is used as features for the CRF. The CRF combines the BiLSTM outputs with a state transition table of possible tags to efficiently use past and future tags to predict the current tag. The Viterbi algorithm [37] is used to efficiently compute optimal tag sequences.

The trainable model parameters include the BiLSTM parameters, the linear layer weights, and the state transition matrix of the CRF. We empirically fix the hyperparameters of the model. We use 200 feed forward units, set the word embedding size to 300, and use a single layer bi-directional LSTM to encode sequences. To see if training time can be reduced, we consider using pretrained word embeddings and training embeddings from scratch. Furthermore, we experiment by enabling and disabling fine-tuning of input embeddings when using pretrained word embeddings. We use negative log-likelihood as the loss function. We train and evaluate the model using 5-fold cross-validation.

5.2. Claim structuring

For claim structuring, we consider two basic approaches: (1) predicting components of the structure and then putting them together to form a claim structure (binary relevance, `BR`) and (2) predicting the entire claim structure at once (label powerset, `LP`). Using components corresponds to a more realistic scenario, as it more faithfully reflects the process of manually annotating claims. Following the `BR` approach, a claim structure can be broken down into four components: approach, a claim structure can be broken down into four components:

- *type* ($TYP \in \{fact, good\_value, bad\_value, policy\}$)
- *arity* ($AR \in \{unary, binary, ternary\}$)
- *relations* ($RE \in \{has\_antecedent, has\_declaration, implies, \ldots\}$)
- *domain concepts* ($DC \in \{marijuana, legalized\_marijuana, mafia\_bankrupt, \ldots\}$)

To construct a claim structure, we use a exactly one (out of four options) *type*, exactly one (out of three options) *arity*, one or more (up to three, out of 22 possible) *relations*, and one or more (one for each relation, out of 82 possible) *domain concepts*. We could have had negation as an extra component of the claim structure, but since only relations may be negated, we construct the relation set *RE* as a union of relations and their respective negations:

$$RE = \{has\_antecedent, negated\_has\_antecedent, \\ implies, negated\_implies, \ldots\}$$

Taking into consideration all possible combinations of components from annotator A1, 107 binary labels $(82 \times DC + 17 \times RE + 4 \times AR + 7 \times TYP)$ can be assigned to a claim, from which a claim structure can be constructed. This entails an exponential (2^{107}) number of possible structures, a large number of which is invalid. An invalid structure would involve both the `has_declaration` and `has_antecedent` relation which is not allowed, since `has_declaration` can only be assigned to a `unary` claim, whereas `has_antecedent` can only be assigned to `binary` claim.

We consider three claim structuring models: a set of independent SVM classifiers (which we use as a baseline), chain classification, and ensemble chain classifiers.

5.2.1. Independent SVMs

As a baseline approach, we use a set of independent SVMs (IND) with distributed word representations as features. To train each independent model, we use 5×3 nested-cross-validation and optimize hyperparameters C and γ using grid search. We experiment with both the `BR` and `LP` approach.

5.2.2. Chain classification

The chain classification (CC) model leverages dependencies among structure components. To first verify the label dependency assumption, we build a chain classifier which uses gold labels as input in each prediction. This yields an overall performance of 0.23 averaged F1-score. The domain concepts proved the hardest to predict, and removing them yields an averaged F1-score of 0.71. We deem this performance to be promising.

We frame all component classifications as either multiclass or multi-label classification. We prefer to use multiclass classification where possible (for type and arity prediction), since generalizing multiclass to multi-label classification usually degrades performance due to loss of information across classes. SVMs models are then chained so that the prediction of one SVM is added as input to the following SVM model, until all labels are predicted. We randomize the ordering of labels which are predicted. Additionally, to alleviate the influence of randomization, we ensemble chain classifiers (ECC) using a majority vote. To train the model, we use 5×3 nested-cross-validation and optimize hyperparameters C and γ using grid search. It is sensible to use the chain classification model only in the `BR` setup.

5.3. End-to-End argumentative claim parsing

The last model we consider is an end-to-end model that jointly performs claim segmentation and clustering. We draw inspiration from [38], where they use consecutive LSTM cells to predict multiple outputs. We adopt a similar approach, where we use two sets of BiLSTM layers: the first layer (`BiLSTM-seg`) to predict segments and the second layer (`BiLSTM-struc`) to infer claim structures of the predicted segments.

The input of the `BiLSTM-seg` layer is a sequence of tokens and their respective part-of-speech (POS) tags [39] paired with `BIO` tags. The `BiLSTM-seg` layer produces `BIO` tags, which are then converted to textual claims. Next, the `BiLSTM-struc` part predicts the claim structures corresponding to these claims. Both the `BiLSTM-seg` and `BiLSTM-struc` employ one BiLSTM and one hidden layer. Softmax is applied to the multiclass type and arity classifiers, whereas the sigmoid function is applied to the multilabel to the domain individual and relation classifiers in order to obtain label probabilities. The output of the `BiLSTM-struc` are class probabilities for all four claim structure components. The model architecture is illustrated in Figure 2. We use three sets of embeddings: word, POS embeddings, and `BIO` tag embeddings. We optimize using the stochastic gradient descent algorithm with a learning rate of 0.01 and use L2 penalty for regularization. Negative log-likelihood loss is used to calculate the loss for `BIO` tags, type, and arity. Binary cross-entropy is used on a per-label basis to calculate loss for relations and domain concepts, since they are predicted in a multi-label fashion. The loss of the joint model is simply the sum of all individual losses. We experiment with and without pretrained word embeddings. In the `LP` setup, we encode each structure as a separate class and then calculate negative log-likelihood.

6. Experiments

6.1. Claim segmentation

For the evaluation, we adopt two sets of metrics: exact and lenient. The first are standard information retrieval

Table 2. Binary relevance multi-label (BR) and `BIO` labelling of comment with two segments: "Nothing can bring peace to this world", "Its a great idea", and one non-argumentative token.

	Nothing	can	bring	peace	to	this	world	Hmmm	Its	a	great	idea
`BIO`	B	I	I	I	I	I	I	O	B	I	I	I
BR	1	1	1	1	1	1	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	1	1	1

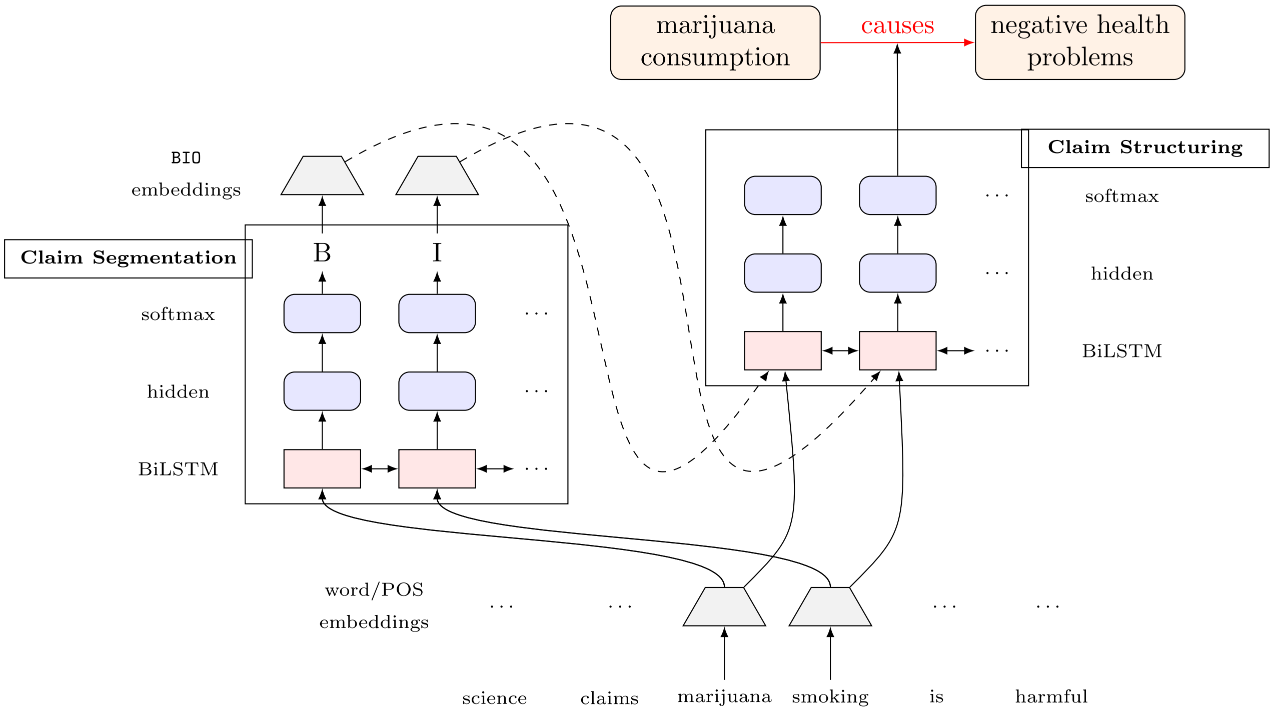

Figure 2. Joint BiLSTM model for both claim segmentation and claim structuring.

metrics of precision (P), recall (R), and F1-score (F1) [40], in which the extracted segment has to match perfectly the annotated segment to be considered a match. The second set of metrics is designed to allow for imperfect matches between extracted and gold segments: lenient precision (l-P), lenient recall (l-R), and lenient F1-score (l-F1). We allow the difference from extracted to gold segments to be up to two tokens in the lenient case. The lenient evaluation metrics are motivated by the assumption that having even imperfectly segmented claims may be sufficient for some ACP applications.

First, we wish to compare the three proposed approaches: the naïve heuristics, the SVM, and the BiLSTM-CRF. For both the SVM and BiLSTM-CRF approach, we use the BR encoding. In the CRF, we do not share the tag transition table across claim segments and train embeddings from scratch. Results are shown in Table 3, with the best results in boldface. The naïve heuristics considerably outperforms both the BiLSTM-CRF and the SVM models that use BR encoding. The heuristics achieves a high precision and low recall, as it favours longer claims segments (that map to sentences). Longer sentences contain more claims, on which the heuristics performed poorly. By inspecting the SVM and BiLSTM-CRF model outputs, we observe mostly majority class predictions, which means those models prefer short comments. The results clearly suggest that models using the BR encoding struggle to predict claim segments.

The lower half of the table shows the results of the models in BIO setup. Since BIO tags cannot be applied to discontinuous and overlapping segments, we remove those claims from the training set. This simplification lowers the upper bound on recall and macro-averaged F1-scores to 0.80 and 0.90, respectively (the upper bound on precision remains 1.0). The results are now more comparable to the naïve heuristics baseline. The BiLSTM-CRF showed the best performance, narrowly outperforming the naïve heuristics by 0.05 and 0.10 F1-score percentage points.

Table 3. Claim segmentation precision (P), recall (R), macro-averaged F1-score (F1), and their lenient counterparts (l-P, l-R, and l-F1) for the BR and BIO setups. The best performing models are in italics.

		Exact			Lenient		
Tagset	Model	P	R	F1	l-P	l-R	l-F1
BR	Heuristic	0.42	0.26	0.32	0.89	0.56	0.69
	SVM	0.03	0.01	0.02	0.09	0.01	0.02
	BiLSTM-CRF	0.12	0.01	0.02	0.32	0.04	0.06
BIO	SVM	0.32	0.16	0.21	0.68	0.33	0.45
	BiLSTM-CRF	*0.43*	*0.32*	*0.37*	*0.93*	*0.69*	*0.79*
	BiLSTM-J[a]	0.32	0.30	0.31	0.69	0.63	0.66

[a] BiLSTM-J is evaluated only on the data from the "Marijuana Legalization" topic

Looking at the results in general, we conclude that predicting the exact boundary of a claim segment is a difficult task, with the best-performing model achieving an F1-score of 0.37. However, the segments produced were often very close to the ones annotated: the lenient macro-averaged F1-score for the best-performing model is 0.79. Inspecting the predicted labels of all models, we observe that none of the models correctly label non-argumentative segments (O labels). The most likely reason lies in the fact that non-argumentative content is greatly underrepresented in this dataset; a possible way to mitigate this would be to use weighted loss or oversampling. Overall, the results

Table 4. Macro-averaged F1-score comparison between the independent SVM (IND), randomized classifier (RAND), chain classification (CC), ensemble chain classification (ECC), and joint model (BiLSTM-J) in the task of claim structuring.

Model / Setup	TYP	AR	RE	DC	CL
RAND	0.25	0.32	0.10	0.04	0.00
IND-BR	*0.84*	*0.79*	0.11	*0.05*	0.00
IND-LP	–	–	–	–	0.08
CC	*0.84*	*0.79*	0.14	0.03	0.02
ECC	0.80	0.77	*0.15*	0.04	0.03
BiLSTM-J	0.52	0.33	0.11	0.00	0.00

Notes: All approaches use BR encoding, except IND-LP which uses the LP encoding. Results are compared on individual components across types (TYP), arity (CNT), relations (RE), and domain concepts (DC). The right-most column (CL) shows macro-averaged F1 for entire claim structures. Best results for each setup are shown in italics.

of claim segmentation are promising, especially when considering the lenient metrics.

6.2. Claim structuring

Evaluation of claim structuring models is carried out in two sets of experiments: in the first, we evaluate models that predict the claim structure on a per-component basis (BR), while in the second, we compare the per-component approach (BR) to label powerset (LP) approach (cf. Section 5.2). Our focus here is on the BR approach, which we deem more realistic, especially in the scenario where the number of domain concepts increases in time.

Table 4 shows macro-averaged F1-scores for the random classifier (RAND), independent SVM classifier (IND), chain classifier (CC), and ensemble chain classifier (ECC) approaches. Models are trained to predict structure components, with their predictions then assembled together to constitute a claim structure, which is compared against the annotated claim structure. The baseline is set using a randomized class picker for all individual components.

Based on the results obtained, we conclude that recognizing the entire claim structure is an extremely challenging task, as the best-performing model (ECC) manages to achieve only 0.03 majority F1-score. This is mainly due to the low performance of recognizing domain concepts in a claim. This is expectedly difficult, as there are 75 sparsely distributed domain concepts, and only 847 claims. Inspecting the component outputs of the independent classifier, we conclude that it mostly manages to correctly classify up to two components (mostly *TYP* and *AR* labels), but never manages to identify three or all four components correctly, hence it never outputs the correct claim structure. Unlike for the IND setup, the CC and ECC models managed to produce fully accurate structures, and correctly predict three out of four component parts for roughly 25% of cases. Even though the CC and ECC models exhibit performance drops in recognizing *TYP* and *AR* labels compared to individual classifiers, overall they seem like the most promising options to structure claims.

We next compare the more realistic BR setup to the less realistic LP setup. For the LP setup, we map each claim structure to a label and employ a single multiclass classifier to predict the claim structure from the claim text. To obtain LP classes, first we generate all possible combinations of structure components yielding 2^{107} possible combinations. Then we restrict the space of possible solutions to only feasible ones resulting in the final 10 652 784 classes, with only 384 classes occurring in the dataset used. The IND-LP model outperforms all BR-based models achieving 0.08 macro-averaged F1-score. We conclude that using the LP setup gives better results than the BR setup. However, due to the size of the dataset and the number of potential classes, we expect that model performance in the LP setup would degenerate for larger and more diverse datasets.

6.3. Argumentative claim parsing

Finally, we evaluate the joint approach to claim structuring and claim segmentation on the "Marijuana Legalization" topic. The joint model might not produce the same number of claim structures per comment, so we evaluate at comment level and average across all comments.

As shown in Table 4, the BiLSTM-J model performs worse than the IND, CC, and ECC models, but better than the random baseline in three out of four categories. This indicates that jointly extracting claims and structuring claims is an extremely difficult task. The joint model fails to successfully extract and structure a single claim. However, as results in Table 3 suggest, the BiLSTM-J does relatively well in the claim segmentation task, achieving comparable results to the BiLSTM-CRF model.

In an attempt to improve the joint model, we experiment with setting lower learning rates for the shared BiLSTM-seg layer, but to no significant performance boost. We then inspect the biggest contributors to the loss during model training and manually assign weights to prevent a single component prediction from dominating the total loss score. By weighting the loss, we do obtain slightly better performance, but consider this to be a short-term workaround.

7. Conclusion

This article introduced *argumentative claim parsing* (ACP), a novel natural language processing task of automatically extracting semantic structures of claims from argumentative text. The task was broken down into two subtasks: claim segmentation and claim structuring. Claim segmentation was formulated as a supervised sequence classification problem, while claim structuring was framed as multi-label classification. We proposed models to tackle claim segmentation and claim structuring separately, and a joint model to solve them jointly. We also described a new dataset, manually annotated for claim segments and claim structures, which we hope will spur further research on this task.

Our experiments reveal that claim segmentation is a difficult, albeit solvable task, yielding 0.37 and 0.79 macro-averaged exact and lenient match F1-score for the best-performing model. Claim structuring proved a much more challenging task, and the best-performing model achieved only 0.03 of majority F1-score. Joint approaches did not yield satisfactory results, especially for the claim structuring problem.

While some of our results are promising, future work should focus on experimenting with alternative models. For claim segmentation, the naöve heuristics produced decent results, so one promising direction might be to expand on similar rule-based approaches. To improve on claim structuring, a sensible stating point would be to constrain the search space to valid structures only, e.g. using linear programming. Furthermore, to ensure reproducibility of our findings, the dataset would have to be extended to cover more topics and more comments per topic. Another promising research direction is the automatic induction of domain concepts from text, building on NLP work in concept extraction and taxonomy induction. Except improving on ACP, we wish to explore how ACP can be used to help solve other related argumentation mining tasks, such as stance classification [41] and argument recognition [42].

Notes

1. https://fasttext.cc/
2. https://code.google.com/archive/p/word2vec/

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Fogg B. Mass interpersonal persuasion: an early view of a new phenomenon. In: International Conference on Persuasive Technology; Springer; Berlin, 2008. p. 23–34.

[2] Petty RE. Attitudes and persuasion: classic and contemporary approaches. New York: Routledge; 2018.

[3] Pang B, Lee L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrievall. 2008;2(12):1–135.

[4] Hasan KS, Ng V. Stance classification of ideological debates: data, models, features, and constraints. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing; 2013. p. 1348–1356.

[5] Jaffe A. Stance: sociolinguistic perspectives. Oxford: OUP; 2009.

[6] Van Eemeren FH, Grootendorst R, Johnson RH. Fundamentals of argumentation theory: a handbook of historical backgrounds and contemporary developments. New York: Routledge; 2013.

[7] Govier T. A practical study of argument. Boston: Cengage Learning; 2013.

[8] Rieke RD, Sillars MO, Peterson TR. Argumentation and critical decision making. Longman; New York; 1997.

[9] Lawrence J, Reed C. Argument mining: a survey. Computational Linguistics. 2020;45(4):765–818.

[10] Lippi M, Torroni P. Argument mining: a machine learning perspective. In: International Workshop on Theory and Applications of Formal Argumentation; Springer; Cham, 2015. p. 163–176.

[11] Freeman K, Farley AM. A model of argumentation and its application to legal reasoning. Artificial Intelligence and Law. 1996;4(3–4):163–197.

[12] Peldszus A, Stede M. From argument diagrams to argumentation mining in texts: a survey. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI). 2013;7(1):1–31.

[13] Palau RM, Moens MF. Argumentation Mining: the detection, classification and structure of arguments in text. In: Proceedings of the 12th International Conference on artificial Intelligence and Law; ACM; 2009. p. 98–107.

[14] Levy R, Bilu Y, Hershcovich D, et al. Context dependent claim detection. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers; 2014. p. 1489–1500.

[15] Sardianos C, Katakis IM, Petasis G, et al. Argument extraction from news. In: Proceedings of the 2nd Workshop on Argumentation Mining; 2015. p. 56–66.

[16] Habernal I, Eckle-Kohler J, Gurevych I. Argumentation mining on the web from information seeking perspective. In: ArgNLP; 2014.

[17] Stab C, Gurevych I. Identifying argumentative discourse structures in persuasive essays. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 46–56.

[18] Hashimoto C, Torisawa K, De Saeger S, et al. Excitatory or Inhibitory: A New Semantic Orientation Extracts Contradiction and Causality from the Web. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; Association for Computational Linguistics; 2012. p. 619–630.

[19] Boltužić F, Šnajder J. Toward Stance Classification based on Claim Microstructures. In: 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis; 2017.

[20] Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. San Francisco, CA. Morgan Kaufmann Publishers Inc.; 2001. p. 282–289; ICML '01. Available from: http://dl.acm.org/citation.cfm?id = 645530.655813.

[21] Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Computation. 2000;12(10):2451–2471.

[22] Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification. In: International Conference on Artificial Intelligence and Soft Computing; Springer; Cham, 2017. p. 553–562.

[23] Ma X, Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv: 160301354.2016.

[24] Reimers N, Gurevych I. Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. arXiv preprint arXiv:170706799. 2017.

[25] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems; 2013. p. 3111–3119.

[26] Mikolov T, Grave E, Bojanowski P, et al. Advances in pre-training distributed word representations. arXiv preprint arXiv:171209405. 2017.

[27] Read J, Pfahringer B, Holmes G. Classifier chains for multi-label classification. Machine learning. 2011; 85(3):333.

[28] Goncalves EC, Plastino A, Freitas AA. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence. IEEE; 2013. p. 469–476.

[29] Hollihan TA, Baaske KT. Arguments and arguing: the products and process of human decision making. Long Grove (illinois): Waveland Press; 2015.

[30] Miller E. An introduction to the resource description framework. Bulletin of the American Society for Information Science and Technology. 1998;25(1):15–19.

[31] Hasan KS, Ng V. Why are you taking this stance? identifying and classifying reasons in ideological debates. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 751–762.

[32] Suykens JA, Vandewalle J. Least squares support vector machine classifiers. Neural processing letters. 1999;9(3):293–300.

[33] Chang CC, Lin CJ. Libsvm: a library for support vector machines. ACM transactionson intelligent systems and technology (TIST). 2011;2(3):1–27.

[34] Habernal I, Gurevych I. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional LSTM. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2016. p. 1589–1599.

[35] Zhou P, Qi Z, Zheng S, et al. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. arXiv preprint arXiv: 161106639.2016.

[36] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:150801991. 2015.

[37] Forney GD. The Viterbi algorithm. Proceedings of the IEEE. 1973;61(3):268–278.

[38] Miwa M, Bansal M. End-to-end relation extraction using LSTMs on sequences and tree structures. arXiv preprint arXiv:160100770. 2016.

[39] Brown RW. Linguistic determinism and the part of speech. The Journal of Abnormal and Social Psychology. 1957;55(1):1.

[40] Hripcsak G, Rothschild AS. Agreement, the F-measure, and reliability in information retrieval. Journal of the American Medical Informatics Association. 2005;12(3): 296–298.

[41] Walker MA, Anand P, Abbott R, et al. Stance classification using dialogic properties of persuasion. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for computational linguistics: Human Language Technologies; Association for Computational Linguistics; 2012. p. 592–596.

[42] Boltužić F, Šnajder J. Back up your stance: recognizing arguments in online discussions. In: Proceedings of the First Workshop on Argumentation Mining; 2014. p. 49–58.

[43] Smith Noah A. Linguistic structure prediction. Synthesis lectures on human language technologies. Vol. 4. Stroudsburg (PA): Morgan & Claypool Publishers; 2011. p. 1–274.