Computational Argumentation — Part IV

Argument Mining

Henning Wachsmuth

https://ai.uni-hannover.de

Natural Language Processing

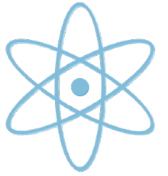
Leibniz Universität Hannover

Learning goals

- **Concepts**
  - Definitions, goals, and tasks in argument mining

- **Methods**
  - Segmentation of argumentative discourse units
  - Classification of types of units
  - Identification of relations between units and arguments
  - Methods that tackle multiple mining tasks jointly

- **Associated research fields**
  - Natural language processing



- **Within this course**
  - The first of three main stages in computational argumentation

Outline: Introduction

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

VII. Applications of computational argumentation

VIII. Conclusion

a) Introduction

b) Unit segmentation

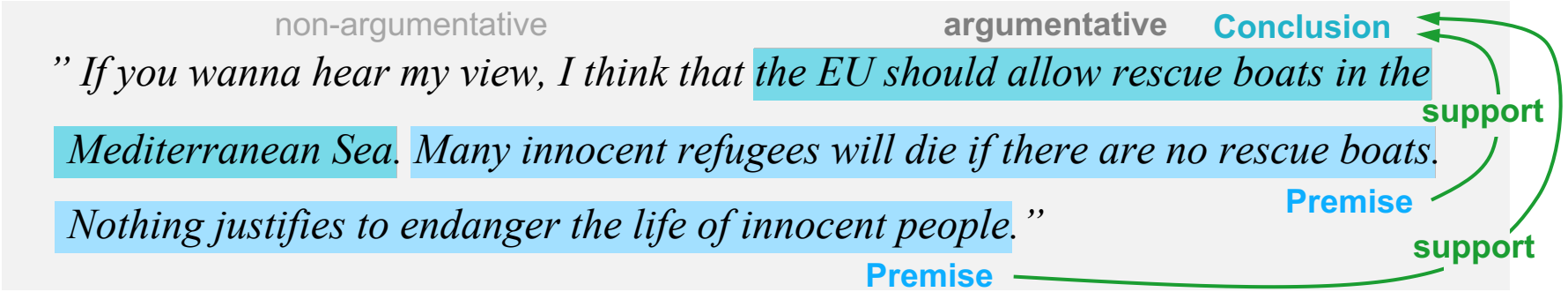
c) Unit type classification

d) Relation identification

e) Conclusion

Argument mining

- **Argument mining** (aka argumentation mining)
  - The identification of argumentative structure in natural language text, in terms of units and their relations
  - May be based on different argument models
  - Often, the argument mining process includes multiple steps.



- **Why argument mining?**
  - Real-world arguments are often “hidden” in longer text, possibly fragmented.
  - Mining provides the basis for any argument analysis and application.

    Exception: Arguments, and their structure, are already given in the source data.

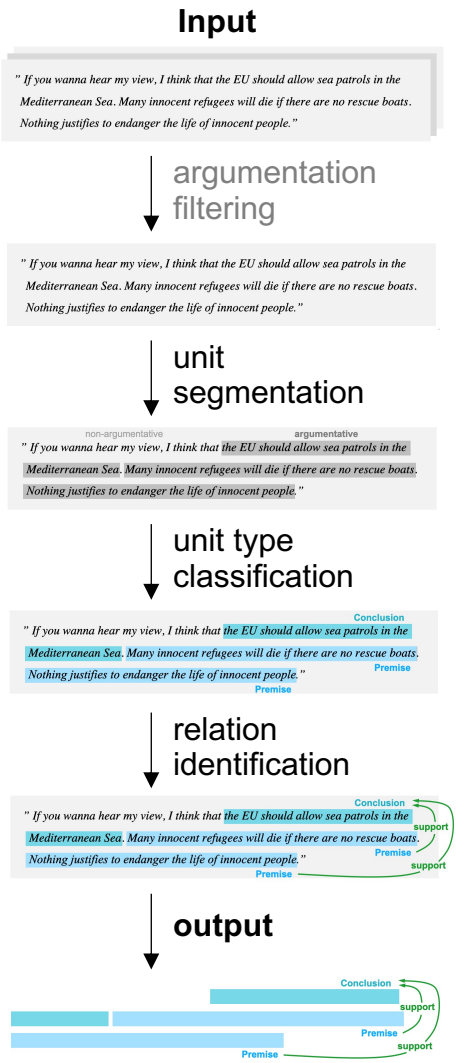
Argument mining: Process

- **General process signature**
  - Input. A set of (plain) texts
  - Output. The argumentative structure of each text

    What structure is mined exactly depends on the argument model employed.

- **Main high-level tasks**
  - Argumentation filtering. Finding argumentative texts
  - Unit segmentation. Finding argumentative units
  - Unit type classification. Deciding types of units
  - Relation identification. Detecting relations between units

- **Notice**
  - Different task decompositions and orderings exist.
  - Some tasks may be tackled jointly, as we see below.
  - Not all tasks always need to be tackled.



Outline: Unit segmentation

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

VII. Applications of computational argumentation

VIII. Conclusion

a) Introduction

b) Unit segmentation

c) Unit type classification

d) Relation identification

e) Conclusion

Unit segmentation

- **Unit segmentation**
  - The segmentation of a text into ADUs, i.e., argumentative units and their non-argumentative counterparts
  - Input. Usually, a plain text (often assumed to be argumentative)
  - Output. All ADUs in the text, defined by their character/token boundaries

non-argumentative **argumentative**

" If you wanna hear my view, I think that the EU should allow rescue boats in the Mediterranean Sea. Many innocent refugees will die if there are no rescue boats. Nothing justifies to endanger the life of innocent people. "

- **Modeling unit segmentation**
  - Individual classification of candidate unit boundaries
  - Sequence labeling of each token in a a given text

    ... along with some variations

Unit segmentation: Example

- **Example: Unit segmentation of essays**
  - Given an essay paragraph on “living overseas”, find all argumentative units.

> *“ Living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet. One who is living overseas will of course struggle with loneliness, living away from family and friends but those difficulties will turn into valuable experiences in the following steps of life. Moreover, the one will learn living without depending on anyone else.”*

example from Stab and Gurevych (2014a)

- **Challenges**
  - What is an argumentative unit may depend on the issue being discussed.
  - Even humans may disagree on the correct segmentation.
  - No clear general definition exists of what makes up the boundaries of units.

    Often, a unit is a clause or sentence w/o discourse markers, but multiple-sentence units exist (Rinott et al., 2015).

Unit segmentation: Genres and approaches

- **Units across genres**
  - Some genres are very dense in terms of argumentative units.
  - Others have a low proportion only, or argumentativeness is issue-dependent.

- **Units in selected genres**
  - Persuasive essays. Nearly everything is argumentative. (Stab and Gurevych, 2014a)
  - News editorials. Many ADUs rather have a *rhetorical* role. (Al-Khatib et al., 2016)
  - Wikipedia articles. Argumentativeness is issue-dependent. (Rinott et al., 2015)
  - Forum discussions. Argumentativeness strongly varies. (Habernal and Gurevych, 2017)

- **Selected approaches to unit segmentation**
  - Rule-based boundary detection using parse trees (Persing and Ng, 2016)
  - Conditional random field (CRF) using diverse features (Stab, 2017)
  - Long short term memory (LSTM) using entity-relation information (Eger et al., 2017)
  - Bi-LSTM using embeddings and diverse features (Ajjour et al., 2017)
  - Bi-LSTM with attention using contextualized embeddings (Spliethöver et al., 2019)

Bi-LSTMs for unit segmentation

- **Unit segmentation as token-level sequence labeling**
  - Given a text, classify each token as belonging to an argumentative unit or not.
  - Each token is beginning (B), inside (I), or outside (O) of a unit.

" If you wanna hear my view I think that the death penalty should be abolished ."

O O O O O O O O O B I I I I I O

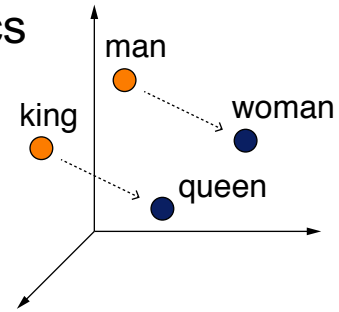
- **Research questions**
  1. What model is best to capture relevant context of a token?
  2. What features are most effective in unit segmentation?
  3. To what extent do models and features generalize across genres?

" If you wanna hear my view I think that the death penalty should be abolished ."

- **Presented approach** (Ajjour et al., 2017)
  - A neural architecture where Bi-LSTMs capture the entire text as context
  - Use of embeddings along with different types of features

Background: Word embeddings

- **Word embedding** (aka word vector)
  - A real-valued vector that represents the distributional semantics of a particular word in a high-dimensional space



$$queen \rightarrow \mathbf{v}_{queen} = (0.13, 0.02, 0.1, 0.4, \ldots, 0.22)$$

  - Words that occur in similar contexts have similar embeddings.
    In other words, similarity can be observed even when different words are used.

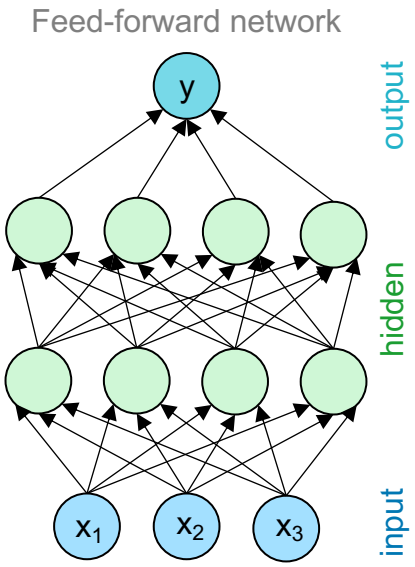
- **Word embedding model**
  - A function that maps each known word to its embedding.
  - Derived from a language model, trained on a (usually huge) corpus

The monarchy is ruled by the _____.

  - Several pretrained embedding models can be found on the web.
    Examples: GloVe, word2vec, Fasttext, Flair, BERT, DeBERTa, …
  - Many embedding models can also be fine-tuned on a given task.

Background: Neural networks

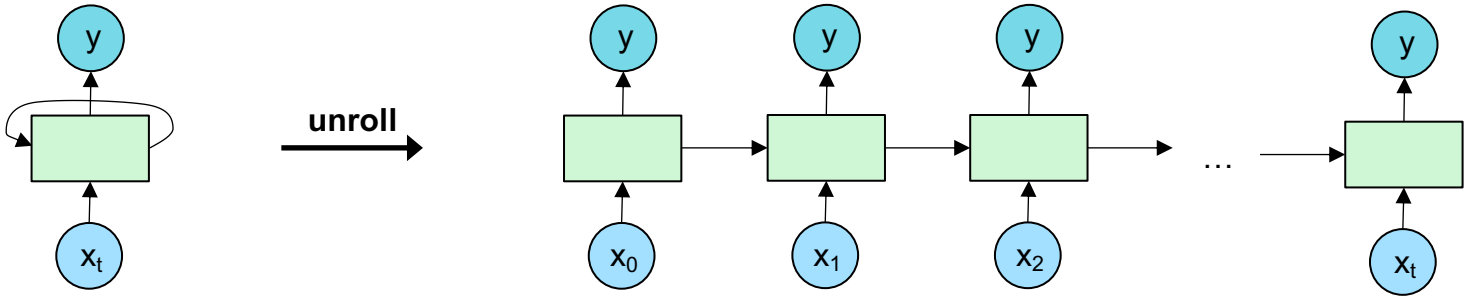
- **Neural network**
  - A layered machine learning model that takes a set of input values and computes one or more output values.
  - Layer. Composes units that can learn complex functions
  - Unit. Computes non-linear weighted sums of input values
    Applied an activation function (e.g., *tanh)* to the sum, weights learned in training



- **Input in NLP**
  - Tokens are represented in the form of word embeddings.
  - Other, human-defined features can be encoded as *one-hot vectors.*
- **Basic types of neural networks**
  - Feed-forward networks. Used for classification and regression
  - Recurrent networks. Used for sequence labeling and generation
    Later in the course, we see further architectures, such as transformers.
- **Notice**
  - In this course, neural network concepts are detailed only as far as needed.
    For a more technical background on neural networks, see the slides of the course “Stastical NLP”.

Background: Recurrent neural networks (Jurafsky and Martin, 2024)

- **Recurrent neural network (RNN)**
  - A network with cycles in its connections, that is, the value of a unit depends on earlier outputs as an input.



  - A text is processed by presenting one token at a time to the network.
  - The layer from step *i* serves as memory (or context) for decisions in step *j* > *i*.

> *" If you wanna hear my view I think that **the** death penalty should be abolished ."*

- **Limitations of simple RNNs**
  - Unidirectionality. Only past input is considered, not future input.
  - Limited memory. Long-term dependencies are hard to learn.

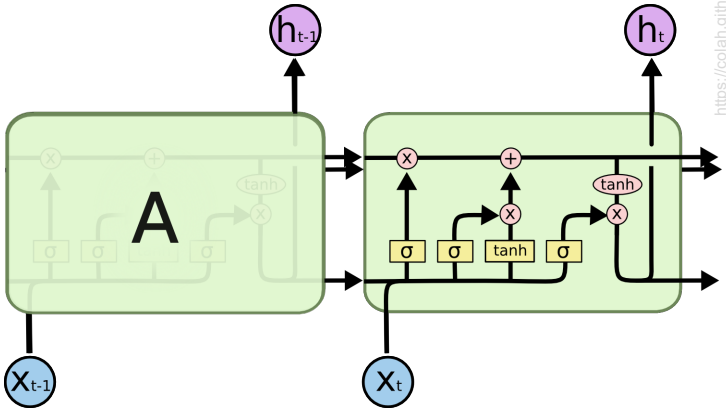
Background: Bi-LSTM neural networks

- **Bidirectional RNN**
  - Two RNNs, one processing a text from left to right, the other from right to left.

    *” If you wanna hear my view I think that **the** death penalty should be abolished .“*

  - The outputs of both RNNs are combined into a single representation.
  - By this, an entire input text can be considered as the context of a token.

- **Long short-term memory (LSTM) unit**
  - Addition of a context layer to a hidden layer that explicitly manages context
  - Three gates learn to decide what context to add, to forget and to use for the output.



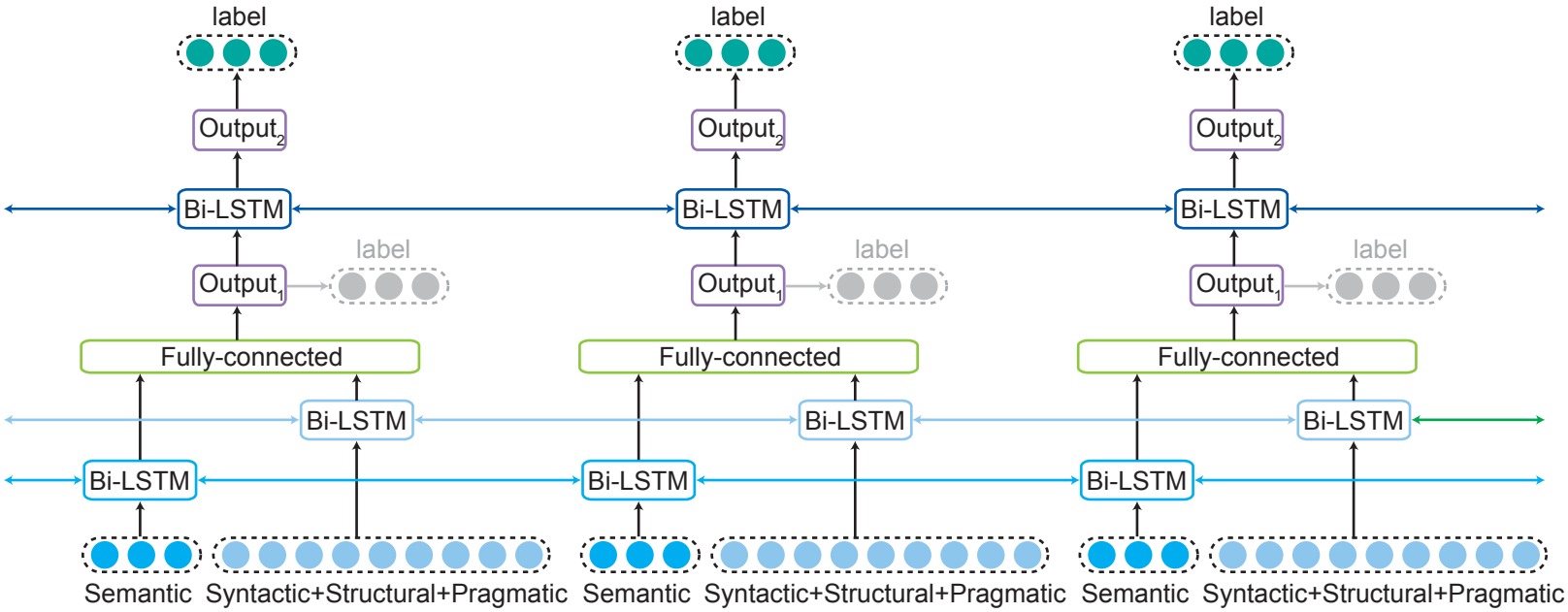
two LSTM units

- **Bi-LSTM**
  - A bidirectional RNN with LSTM units

    Multiple Bi-LSTMs (as well as other neural networks) can easily be stacked.

Bi-LSTMs for unit segmentation: Approach

- **Bi-LSTM-based unit segmentation** (Ajjour et al., 2017)



Architecture illustration for three consecutive tokens

- The first Bi-LSTM layers encode semantic features as word embeddings, others as one-hot vectors.
- Another Bi-LSTM layer models interdependencies of consecutive predictions.
- The output layers predict confidence values for the possible labels (B, I, O).

Bi-LSTMs for unit segmentation: Experiments

- **Baselines**
  - SVM. Linear support vector machine that classifies each token independently
  - CRF. Linear-chain conditional random field that classifies each token in the context of its $k = 5$ surrounding tokens

- **Features**
  - Semantic. The token‘s embedding (for Bi-LSTM) or its text (for SVM, CRF)
  - Structural. If token is at start, inside, or end of a sentence, clause, or phrase
  - Syntactic. Part-of-speech tag of the token
  - Pragmatic. If token is before or after a discourse marker, or in-between two

- **Data**
  - Essays. 402 student essays (Stab, 2017)
  - News. 300 news editorials (Al-Khatib et al., 2016)
  - Web. 340 forum posts, comments, … (Habernal and Gurevych, 2015)

Corpus	B	I	O
Essays	6 089	94 411	44,022
News	14 234	251 381	21 849
Web	1 129	40 042	44 814

Bi-LSTMs for unit segmentation: Results

- **Cross-domain evaluation**
  - Train model on training set (and optimize on validation set) of one genre
  - Apply model to test sets of all three genres

- **Overall results** (token-level macro F_1)

Approach	Test on essays			Test on news editorials			Test on web discourse		
	Essays	News	Web	Essays	News	Web	Essays	News	Web
SVM	61.4	50.9	31.3	58.8	79.9	**22.6**	**39.1**	37.4	42.8
CRF	79.2	52.5	21.7	**69.8**	82.0	8.0	37.1	**37.6**	37.7
Bi-LSTM	**88.5**	**57.1**	**37.0**	60.7	**84.1**	20.9	20.9	36.6	**54.5**

  - 88.5 significantly better at $p < .001$ than best result before (86.7) (Stab, 2017)
  - In general, cross-genre effectiveness limited

- **Feature analysis**
  - Semantic features best in-genre (e.g., 87.9 on essays)
  - Structural features most genre-robust (e.g., 35.5–39.5 on web discourse)

Unit segmentation: Discussion

- **Effective unit segmentation**
  - Diverse approaches to unit segmentation may be considered.
  - High effectiveness seems possible in rather homogeneous genres.
  - The context of tokens is critical to assess their argumentativeness.

- **Definition of units**
  - The exact difference to syntactic and discourse units remains to be studied.
  - Depending on the genre, units can span anything from clauses to paragraphs.
  - To a certain extent, unit segmentation is genre-specific.

- **Knowledge for segmentation**
  - It is debatable whether unit segmentation should be tackled first.
  - At this point, no knowledge is given about what is argued about.
  - Joint mining approaches may often be preferable in practice. (Eger et al., 2017)

Outline: Unit type classification

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

VII. Applications of computational argumentation

VIII. Conclusion

a) Introduction
b) Unit segmentation
c) Unit type classification
d) Relation identification
e) Conclusion

Unit type classification

- **Unit type classification**
  - The assignment of a class to each argumentative unit from a predefined set of classes (that indicate role or other types)
  - Input. A set of argumentative units, often ordered and grouped by input text
  - Output. Each unit with assigned type

Conclusion

" If you wanna hear my view, I think that the EU should allow rescue boats in the Mediterranean Sea. Many innocent refugees will die if there are no rescue boats.

Premise

Nothing justifies to endanger the life of innocent people. "

Premise

- **Modeling unit type classification**
  - Supervised text classification of each unit, either feature-based or neural
  - Sequence labeling on the unit level

    Some approaches also tackle unit types as part of relation identification (more below)

Unit type classification: Example

- **Example: Unit type classification in essays**
  - Given the following essay units, identify their type (conclusion vs. premise).

Conclusion

"*Living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet.* *One who is living overseas will of course struggle with loneliness, living away from family and friends* but *those difficulties will turn into valuable experiences in the following steps of life.* Moreover, *the one will learn living without depending on anyone else.*"

Premise

Premise

Premise

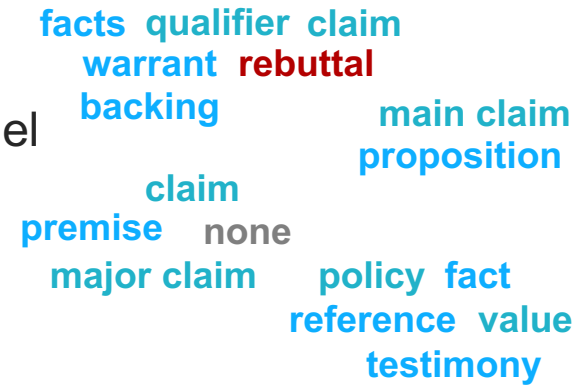
example from Stab and Gurevych (2014a)

- **Challenges**
  - Unit types may be issue-dependent, e.g., whether a unit is evidence.
  - Positional information is not always as helpful as for essays.
  - Some types encode structural information, others semantics or pragmatics.

Unit type classification: Genres and approaches

- **Unit types across genres**
  - Unit types may indicate roles, claim and evidence types, or similar.
  - Unit type schemes are model-specific rather than genre-specific.

- **Selected unit type schemes**
  - Theoretical models. Toulmin model, Freeman model (Toulmin, 1958; Freeman, 2011)
  - NLP models. Essay-specific, Evaluability-oriented (Stab, 2017; Park et al., 2018)

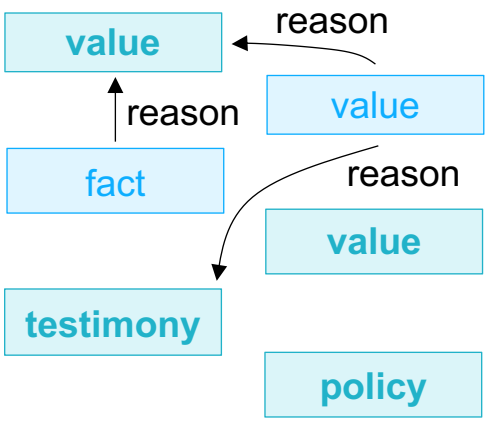


- **Selected approaches to unit type classification**
  - Supervised classification with rich linguistic features (Stab and Gurevych, 2014a; Habernal and Gurevych, 2015; Rinott et al., 2015; Persing and Ng, 2016; Al-Khatib et al., 2017)
  - Unit-level sequence labeling with rich linguistic features (Habernal and Gurevych, 2017)
  - Structure learning for graph prediction with SVMs and RNNs (Niculae et al., 2017)
  - Tree kernels based on syntactic parse trees (Liga, 2019)
  - Biaffine attention for unit-level sequence labeling (Morio et al., 2020)

Biaffine attention for unit type classification

- **Unit type classification in real-world argumentation**
  - Often, writers mix different claims and reasons with partial structure only.
  - Classifying unit types may require knowledge about the units‘ relations.

I‘m with Massachusetts on this one. Repetitive and robo-calls are annoying and not productive. Another fact about robo-calls is that their messages often start in the middle, or maybe this is done on purpose. When it has happened to me, I just hang up. Policies regulating the number of contacts made within a specific time period should include all modes of technology.

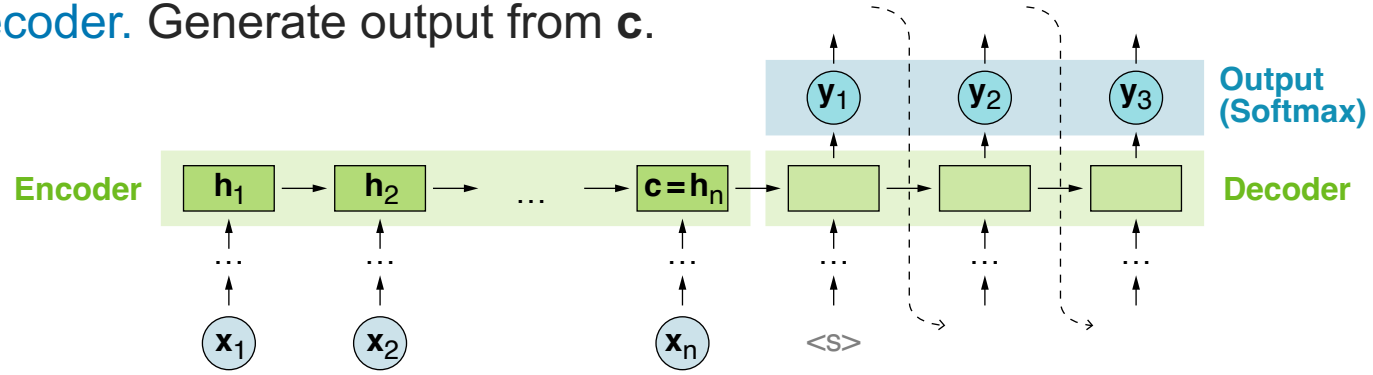


- **Biaffine attention approach** (Morio et al., 2020)
  - Jointly classify unit types and identify relations to model inderdepencies
  - Bi-LSTMs learn to put attention on related pairs of units

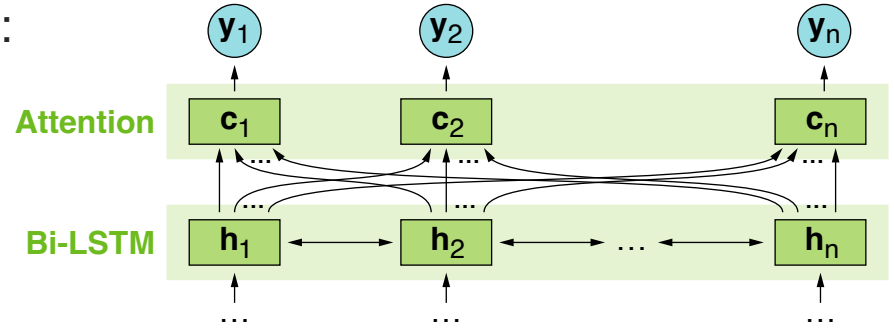
    While both unit types and relations are modeled, the approach could be used for either only.

Background: Attention in RNNs (Jurafsky and Martin, 2023)

- **Encoder-decoder RNN**
  - An RNN that separates input encoding from output decoding
  - Encoder. Process whole input to create a context representation $\mathbf{c} = \mathbf{h}_n$
  - Decoder. Generate output from **c**.

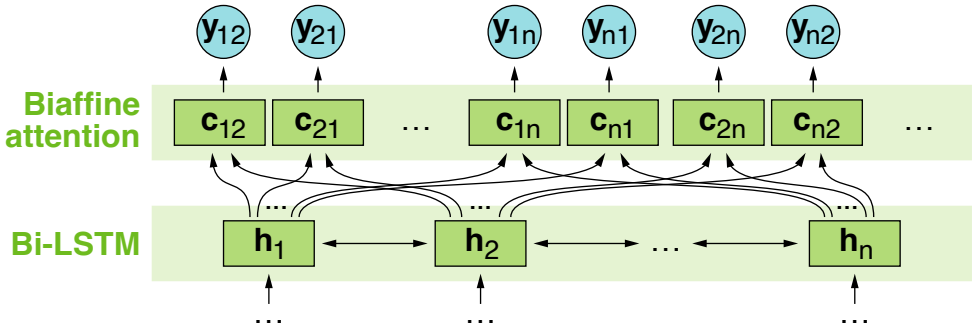


- **Attention**
  - Retain hidden states to learn which inputs are relevant to which outputs.
  - Condition **c** on *all* outputs of layer **h**:
    $\mathbf{c} := f(\mathbf{h}_1 . . . , \mathbf{h}_n)$.
    Often, **h** is modeled as a Bi-LSTM layer.
  - Use separate context $\mathbf{c}_t$ in each decoding step *t*.



Background: Biaffine attention (Dozat and Manning, 2017)

- **Biaffine attention**
  - Represent all possible pairs of inputs (instead of single inputs).
  - Learn the relation of input pairs to outputs.

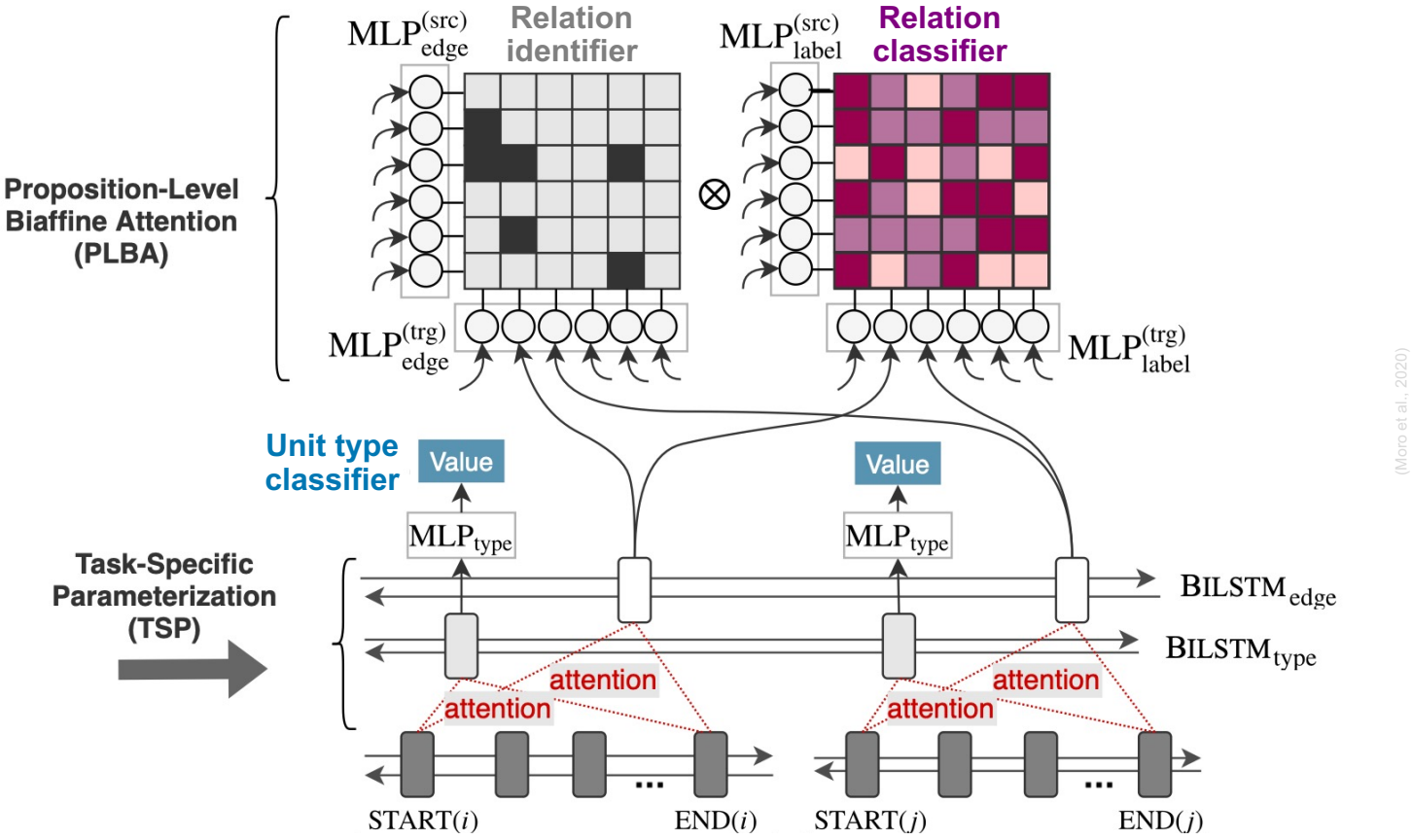


- **Notice**
  - Attention and biaffine attention model input-output connections.
  - This resembles how transformers work, but is not exactly the same.
    Details later in the course

- **Unit-level biaffine attention**
  - Each input is one argumentative unit of the text.
  - Relations may exist between any pair of units that also affect the units‘ types.

Biaffine attention for unit type classification: Approach

- **Biaffine attention for unit type classification** (Morio et al., 2020)
  - Unit-level biaffine attention. Model relations and their types
  - Task-specific parameterization. Use separate attention layer for unit types



Biaffine attention for unit type classification: Results

- **Data** (Park et al., 2018)
  - 731 forum arguments, annotated for evaluability model

policy fact
reference value
testimony

- **Baselines** (Niculae et al., 2017)
  - SVM-based graph prediction. Previous state of the art, jointly predicting entire argument graph structures using SVMs
  - RNN-based graph prediction. Same idea, prediction with RNN

- **Results** (F_1-score)

Approach	Unity types	Relations
SVM-based graph prediction	73.2	26.7
RNN-based graph prediction	72.7	14.4
Biaffine attention	**78.9**	**34.0**

  - Strong improvements for unit types
  - Effectiveness on relation identification task unsatisfying

Unit type classification: Discussion

- **Unit type classification**
  - Early approaches model unit type classification as standard text classification.
  - Some approaches jointly segment and classify units using sequence labeling.
  - More recent approaches classify unit types and relations jointly.

- **Effectiveness of unit type classification**
  - Effectiveness often reasonably high across various genres and models
  - State-of-the-art F_1-scores range from 0.77 (news editorials) to 0.87 (essays) (Stab, 2017; Al-Khatib et al., 2017; Morio et al., 2020)
  - Still, minority unit types may be hard to classify accurately.

- **Unit types as roles?**
  - Conceptually, classifying the argumentative *role* of a unit is questionable, because one unit may have different roles in different arguments.
  - Still, role classification works well in narrow genres, such as essays. Why? Stab (2017) distinguished major claims, claims, premises, and none.

Outline: Relation identification

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

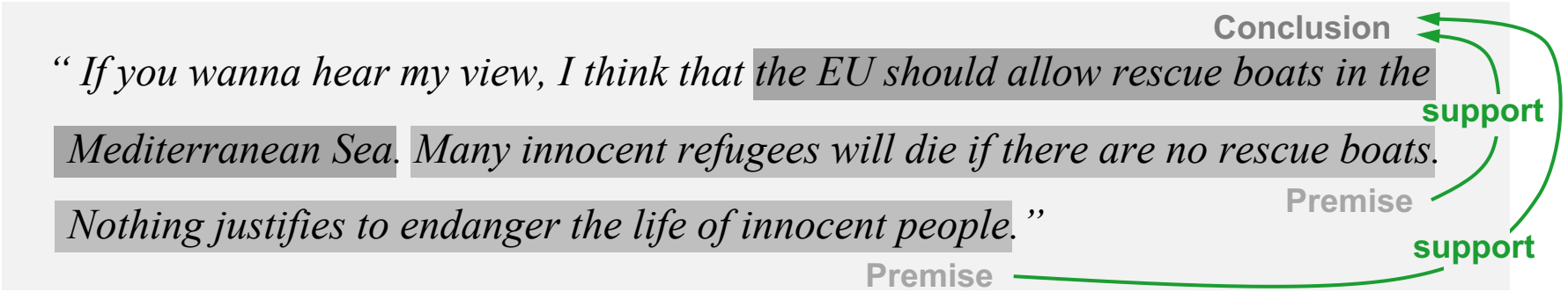
VII. Applications of computational argumentation

VIII. Conclusion

a) Introduction
b) Unit segmentation
c) Unit type classification
d) Relation identification
e) Conclusion

Relation identification

- **Relation identification**
  - The mining of argumentative relations between pairs of units and the classification of the types of relations, usually as *support* or *attack*
  - Input. A set of argumentative units in a text, possibly with assigned unit type
  - Output. All found argumentative relations, with their type

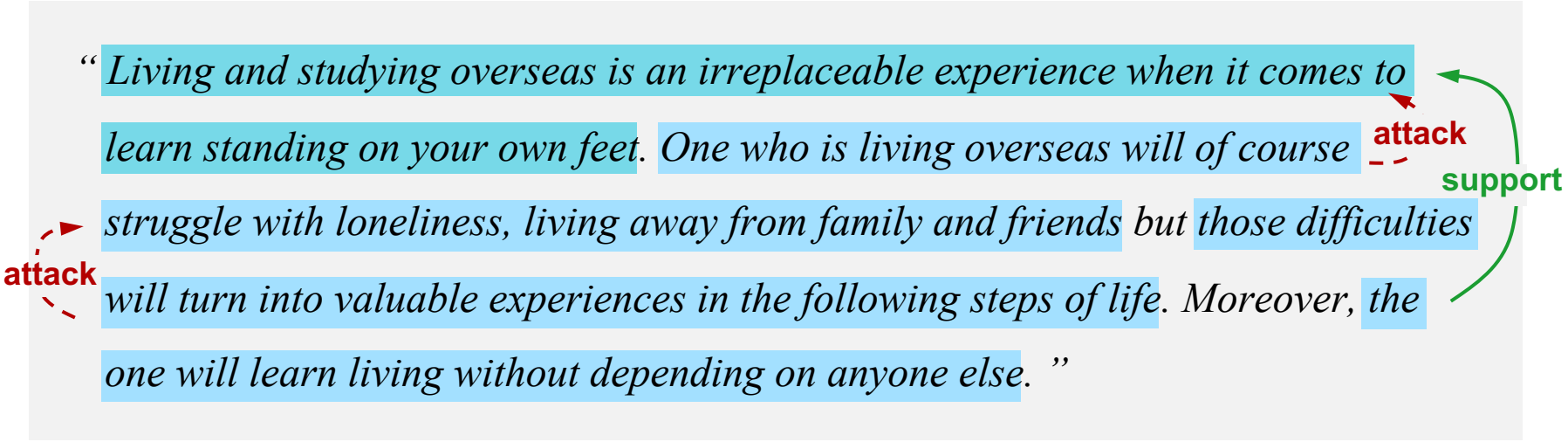


- **Modeling relation identification**
  - Individual classification of candidate unit pairs
  - Identification of the most likely graph induced by all units and relations

    ... among other ways (more below)

Relation identification: Example

- **Example: Relation identification in essays**
  - Given the following essay units, identify all support and attack relations.



example from Stab and Gurevych (2014a)

- **Challenges**
  - Technically, two tasks need to be solved: mining and type classification.
  - In some genres, related units may be far away from each other.
  - Subtle argumentation leaves relations implicit on purpose.

Relation identification: Genres and approaches

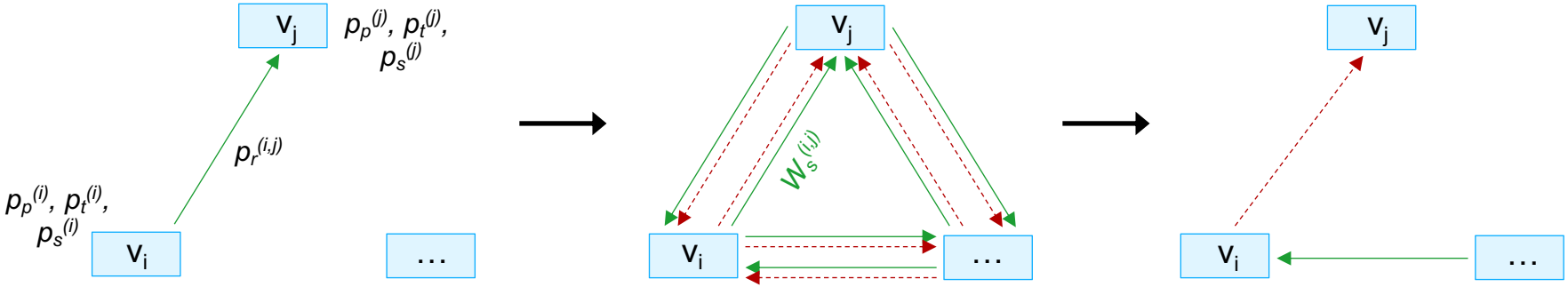
- **Argumentative relations across genres**
  - The idea of support and attack is genre-independent.
  - Some argument models consider different relation sub-types.

- **Selected relation schemes**
  - Essay-specific model. Simple support and attack (Stab and Gurevych, 2014a)
  - Freeman‘s model. Multiple types of support and attack (Peldsdzus and Stede, 2013)
  - Walton‘s model. Inference relations of argument schemes (Lawrence and Reed, 2017)

- **Selected approaches to relation identification**
  - Maximum spanning tree on classified roles and functions (Peldszus and Stede, 2015)
  - Topic modeling based on inferential topic pairs (Lawrence and Reed, 2017)
  - Structure learning for graph prediction with SVMs and RNN (Niculae et al., 2017)
  - Biaffine attention for unit-level sequence labeling (Morio et al., 2020)
  - Transition-based parsing using BERT and LSTMs (Bao et al., 2021)
  - Prompting-based identification with large language models (Gorur et al., 2025)

MST-based relation identification

- **Task**
  - Given the segmented units of a text, classify their types, mine their relations and classify the types of relations.
- **Data** (Peldszus and Stede, 2015)
  - Arg-microtexts. 112 texts with 576 units, in both English and German
  - Annotated for Freeman‘s model, but simplified to (single) support and attack
- **Presented approach** (Peldszus and Stede, 2015)
  - Supervised classification to obtain role and function probabilities
  - Weighted probability aggregation to obtain evidence graph
  - Maximum spanning tree (MST) to obtain relations



MST-based relation identification: Classification

- **Supervised classifiers**
  - Linear log-loss models that predict probabilities of four labels
  - Role ($p_p^{(i)}$). Whether unit *i* is on the proponent or opponent side
  - Thesis ($p_t^{(i)}$). Whether unit *i* is a thesis or not
  - Function ($p_s^{(i)}$). Whether unit *i* has a supporting (or attacking) function
  - Relation($p_r^{(i,j)}$). Whether unit *i* is in relation to unit *j*
- **Employed features**
  - Content and style. Lemma *n*-grams, POS tags, discourse markers, …
  - Structure. Length and position of unit, distance and order of unit pair
- **From node labels to edge labels**

$$p_p^{(i,j)} := \begin{cases} p_p^{(i)} \cdot p_p^{(j)} + (1 - p_p^{(i)}) \cdot (1 - p_p^{(j)}) & \text{if } (i,j) \text{ support edge} \\ p_p^{(i)} \cdot (1 - p_p^{(j)}) + (1 - p_p^{(i)}) \cdot p_p^{(j)} & \text{if } (i,j) \text{ attack edge} \end{cases}$$

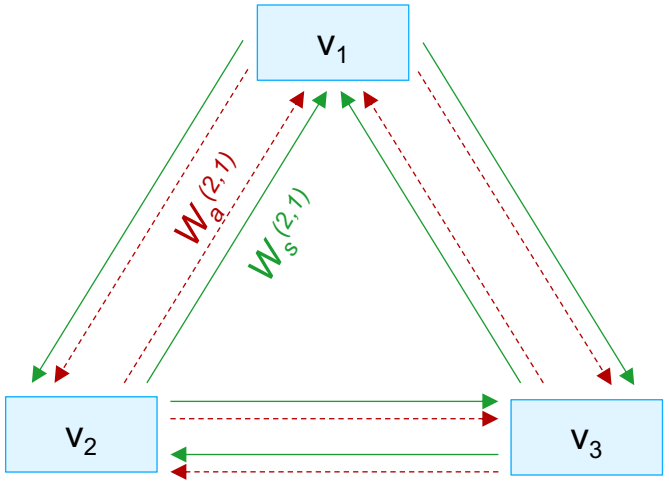
$$p_t^{(i,j)} := 1 - p_t^i \qquad p_s^{(i,j)} := \begin{cases} p_s^{(i)} & \text{if } (i,j) \text{ support edge} \\ 1 - p_s^{(i)} & \text{if } (i,j) \text{ attack edge} \end{cases}$$

MST-based relation identification: Aggregation

- **Weighted probability aggregation**
  - Add weight to each probability of a given candidate unit pair.
  - Learn weights for each probability on training set.

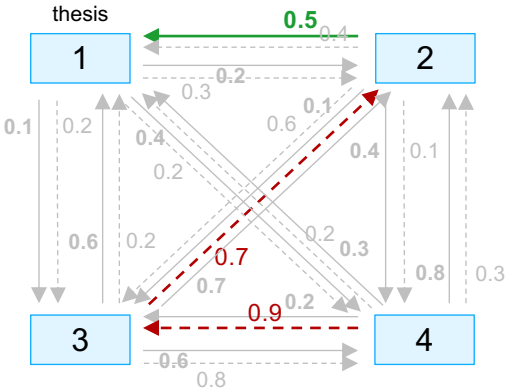
$$w^{(i,j)} = \frac{w_p \cdot p_p^{(i,j)} + w_t \cdot p_t^{(i,j)} + w_s \cdot p_s^{(i,j)} + w_r \cdot p_r^{(i,j)}}{\sum_k w_k}$$

- **Evidence graph**
  - A weighted directed graph *G = (V, E)*
  - Nodes. Each node *v* in *V* represents an ADU.
  - Support edges. Any pair of nodes v_i, v_j is connected with an edge e_s.
  - Attack edges. Any pair of nodes v_i, v_j is connected with an edge e_a.
  - Weights. Each *e* is labeled with a weighted pair score $w^{(i,j)}$ as defined above.



MST-based relation identification: Approach

- **Maximum spanning tree (MST)**
  - A subgraph G^* of a weighted graph $G = (V, E)$ whose edges E connect all nodes V with maximum weight
  - MSTs have $|V|-1$ edges; the root has no outgoing edge.
  - For directed graphs, can be found in $O(|E| + |V| \log |V|)$

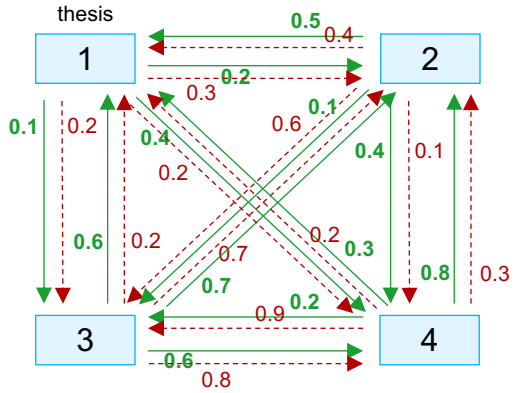


- **Chu-Liu-Edmonds MST** (Chu and Liu, 1965; Edmonds, 1967)
  - Input. A directed weighted graph $G = (V, E)$ with root v_R in V
  - Output. An MST G^* of G

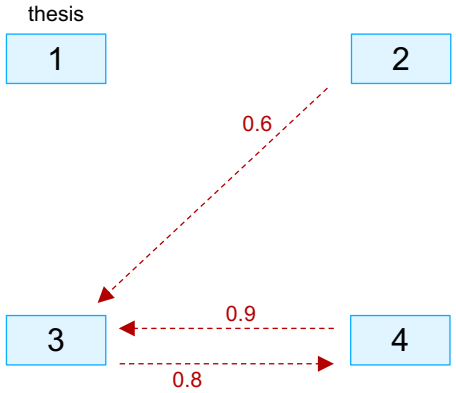
1. Initialize G*: For each node $v \neq v_R$, add edge (v, v') of v with maximum score.
   In case of ties, prefer support edges. Pick randomly from these.
2. For each cycle: Replace edge (v, v') with minimum score by new edge (v, v'') with maximum score, $v' \neq v''$.
3. Repeat Step 2 until no more circle exists.
4. Return G*.

MST-based relation identification: Example

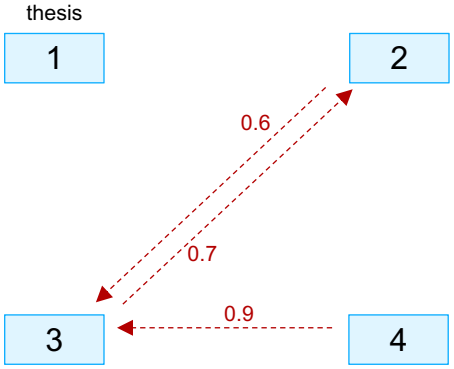
1. Evidence graph G



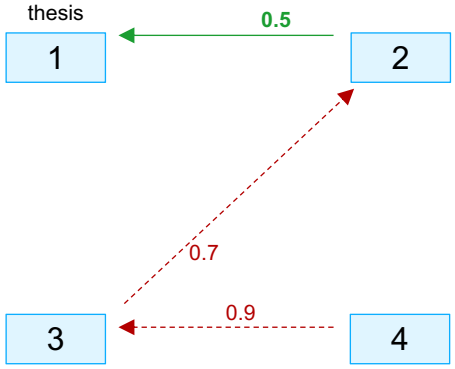
2. Initialize G*



3. Replace edge to remove cycle



4. Replace edge to remove cycle



MST-based relation identification: Results

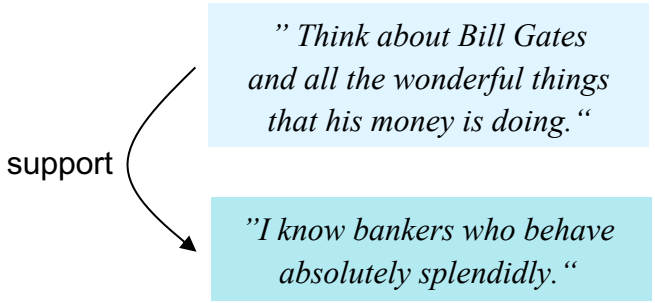
- **Baselines**
  - Classifiers. Determine whether one unit supports or attacks another, or not.
  - Discourse. Fine-tuned discourse parser (instead of evidence-based MST)
  - Classifiers + Discourse. Discourse parser, using classifier outputs as features
- **Results** (macro F_1-score)
  - 5-fold cross validation (10 times repeated), on English and German test set

Approach	Unit	Relation	Rel. type	Unit	Relation	Rel. type
Classifiers	0.82	0.66	0.67	0.85	0.68	0.70
Discourse	0.78	0.71	0.49	0.83	0.72	0.50
Classifiers + Discourse	0.83	**0.72**	0.68	0.86	**0.73**	0.72
Classifiers + MST	**0.87**	0.69	**0.71**	**0.89**	0.71	**0.74**

- **Discussion**
  - Approach best for unit and relation types (not relations) in both languages
  - The MST idea makes sense, if full argumentative structure can be expected.
  - Otherwise, some kind of argument decomposition is needed before.

Topic modeling for relation identification

- **Task**
  - Given two argumentative units, decide if they are in a premise-conclusion relation.



- **Test data**
  - 327 units with 128 relations from transcript of radio show on morals in the banking system

- **Presented approach** (Lawrence and Reed, 2017)
  - Retrieve web pages based on 1- and 2-grams in units.
  - Acquire training data using high-precision indicators.
  - Model topics of the related units.
  - Identify new relations based on topic-pair probabilities.

- **Retrieval of web pages**
  - Retrieve 200 Google results for top-10 1- and 2-grams.
  - This led to 6891 web pages.

1-gram	#
investment	39
banking	35
banks	28
…	…

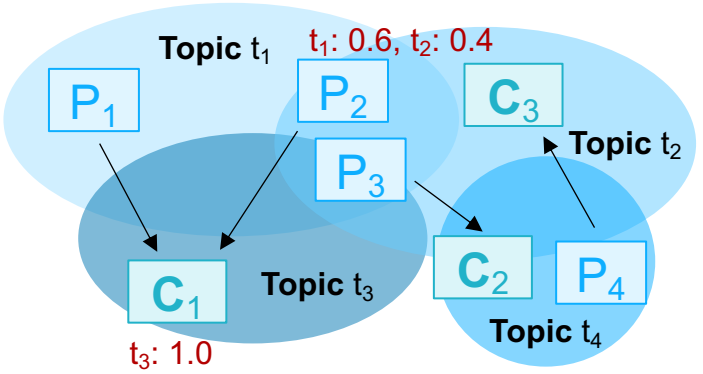
2-gram	#
invest. bank	18
invest. banking	12
common good	5
…	…

Topic modeling for relation identification: Approach

- **Identification of training data**
  - Obtain high-precision discourse markers of relations from sample data.
  - Use top markers for identification (top 2).
  - This led to 7162 training sentences.

Indicator	P	R
P *therefore* C	.95	.0004
C *because* P	.91	.0031
P *consequently* C	.82	.0001
P *hence* C	.76	.0001
...	...	...

- **Modeling topics of related units**
  - Split sentences into two units at markers.
  - Use LDA topic modeling to cluster units into overlapping topics
  - 40 topics were modeled this way.



- **Identification of new relations**
  - Compute probabilities of each specific pair.
  - Derive probabilities of topic pairs in general.
  - Relate units if probability is above average.

P_2 C_1

$p(t_1, t_3) = 0.6 \cdot 1.0$

$p(t_2, t_3) = 0.4 \cdot 1.0$

	t_1	t_2	t_3	t_4
t_1	.43	.42	.11	.04
t_2	.21	.02	.45	.32
t_3	.05	.51	.28	.16
t_4	.31	.05	.16	.48

Topic modeling for relation identification: Results

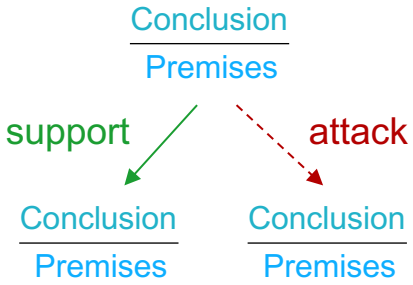
- **Approaches**
  - Random baseline. Classify relations randomly
  - Approach (MaxTopic). Probability of pair with most likely topic of each unit
  - Approach (Weighted). Highest probability across all potential topic pairs
- **Results** (on balanced test set)
  - Directionality. Classify whether two units are in premise-conclusion relation
  - Connectedness. Classify only whether the units are related

Approach	P	R	F_1	P	R	F_1
Random	0.50	0.50	0.50	0.50	0.50	0.50
MaxTopic	0.61	0.77	0.68	0.59	0.75	0.66
Weighted	**0.65**	**0.78**	**0.71**	**0.65**	**0.83**	**0.72**

- **Discussion**
  - Recall is rather decent, but precision remains limited.
  - Still, the approach works fully unsupervised.

Prompting-based relation identification

- **Task**
  - Given two arguments, does the second support or attack the first?



- **Data**
  - 10 existing argument corpora, boiled down to binary attack vs. support
  - Corpora include persuasive essays, forum arguments, Arg-Microtexts, Kialo debate portal arguments, and more

    (Stab, 2017; Park et al., 2018; Peldszus and Stede, 2015)

" ChatGPT will reach AGI level by 2030." attack *" To reach AGI, it should be able to generate its own goals and intentions: where would it draw these from?"*

- **Presented approach** (Gorur et al., 2025)
  - Prompting a general instruction-following large language model (LLM)
  - In-context learning to prime the LLM on the task

Background: Large language models (LLMs)

- **Language model (LM)**
  - Represents a probability distribution over word sequences, derived from data
  - Probabilities can be used to generate most likely *next* words

Prompt: Can you tell me an argument in favor of having rescue boats?

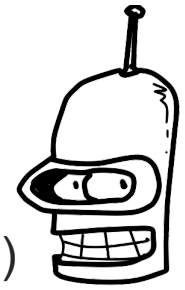
LM: Without such boats, many innocent refugees will <?> →

P("die" | dialogue) = .04
P("drown" | dialogue) = .03
P("suffer" | dialogue) = .01

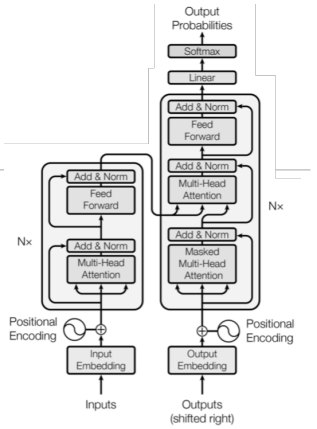
  - Its input representation can also be used for classification and regression.

- **Large language model (LLM)**
  - Large is not exactly defined, but most LLMs have billions of parameters.
  - Mostly, a pretrained transformer LM is meant that follows instructions, that is, it answers to prompts.

- **Types of LLMs**
  - Base. Transformer-based (GPT-3, BART, …)
  - Instruct/Chat. Instruction fine-tuned and aligned (GPT-3.5, LLaMA, …)

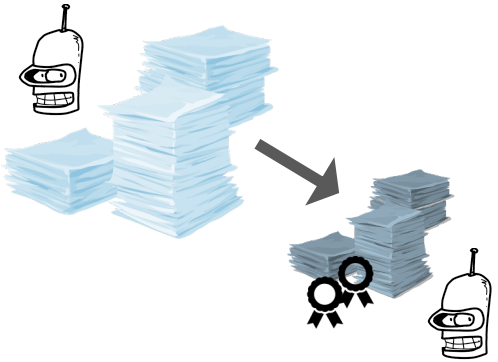


Background: Core concepts of LLMs



Vaswani et al. (2017)

- **Transformer**
  - A neural network architecture for parallel input processing
    The transformer architecture is detailed later in this course.

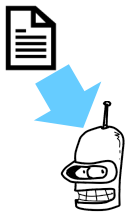


https://en.wikipedia.org

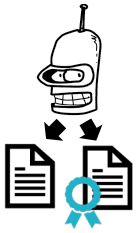
https://letsdraw.it

- **Transfer learning**
  - Pretrain network self-supervised on huge text data.
  - Fine-tune it supervised on task-specific training data.
  - Enables LLMs to leverage knowledge across contexts

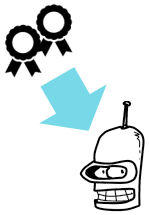
- **Instruction fine-tuning (IFT) and alignment**
  - IFT. Train LLMs to create answer-like output to any instruction
  - Alignment. Optimize answers towards human-defined preferences



1. Instruction fine-tuning.
LLM trained on human answers to prompts



2. Human feedback.
Human rewards answers of LLM



3. Proximal policy optimization.
LLM aligned to human rewards

https://letsdraw.it

- Enables LLMs to give reasonable answers to nearly any prompt

Background: Prompts and few-shot prompting

- **Prompt**
  - Input given to an LLM, serving as context for output generation
  - Prompting. The act of phrasing a prompt to tackle a given task
  - Prompt engineering. The (manual) tuning of prompts to boost effectiveness

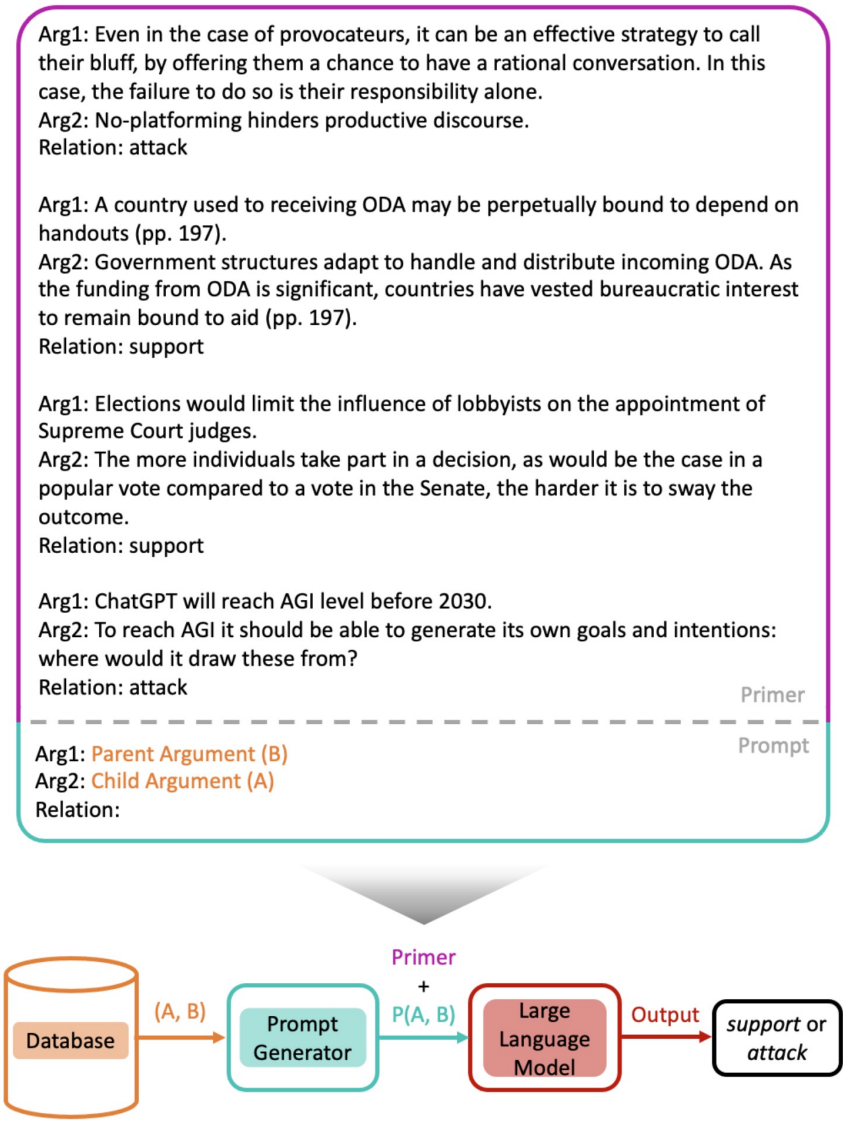
Persona	Imagine you are doing the customer relationship management of a hotel, analyzing what pasts guests think about your hotel.
Task description	You should classify the sentiment polarity of this opinion: <INPUT>
Definition	An opinion is a statement that evaluates a specific aspect of the hotel.
rectional stimulus	You should output one of two label as the polarity: "positive" or "negative".
Reasoning steps	To do so, first identify the aspect being talked about in the statement. Then, identify what sentiment is expressed towards the aspect and decide whether this is positive or negative for the hotel. The resulting label is

- **Few-shot prompting**
  - The inclusion of k ≥ 1 examples of the task (shots) in the prompt
  - This affects the LLM's behavior and how the output looks like.

Shot 1	Opinion: the room was clean and cozy. Polarity: positive
Shot 2	Opinion: this alone never justifies the price. Polarity: negative
	Opinion: <INPUT>. Polarity:

Prompting-based relation identification: Approach

- **Approach** (Gorur et al., 2025)
  - Training data. Start from training set of argument relations.
  - Few-shot prompting. Combine four training examples with new instance.
  - Classification. Let LLM classify new instance as attack or support.
- **Used LLMs**
  - Llama 2. LLM developed by meta; models with 13B and 70B parameters (Touvron et al., 2023)
  - Mistral. LLM by French company; models with 7B and 8x7B parameters (Jiang et al., 2024)



Prompting-based relation identification: Results

- **Baseline**
  - RoBERTa. Transformer-based encoder, fine-tuned on one dataset (Kialo)
- **Results** (macro F1-score)

Approach	Essays	CDCP	Arg-Microtexts	…	Kialo
RoBERTa	0.80	0.75	0.67	…	n/a
Llama 2 (13B)	0.82	0.87	0.67	…	0.65
Llama 2 (70B)	**0.90**	0.92	**0.73**	…	**0.86**
Mistral (7B)	0.85	0.75	0.67	…	0.83
Mistral (8x7B)	0.89	**0.93**	0.70	…	0.84

- **Discussion**
  - The approach is as simple as it gets with LLM prompting.
  - Prompt engineering could further evolve such an approach.
  - The binary decision presupposes that the arguments are in relation.
  - Still, results indicate general impact of LLMs.

Relation identification: Discussion

- **Relation identification**
  - Diverse approaches have been proposed for relation identification.
  - Some focus on *support*, the default relation from premise to conclusion.
  - Most tackle relations between units, but some also between full arguments.

- **Effectiveness of relation identification**
  - Mining relations usually works comparably better than classifying their type.
  - Semi-reliable for explicit argumentation (Park et al., 2018)
  - Unsolved for “hidden” argumentation, even hard for humans (Al-Khatib et al., 2017)

- **Difference to stance**
  - Attack/support and pro/con stance classification conceptually overlap.
  - Unlike relations, stance refers to the author‘s position on an issue.
  - Still, support/attack can be modeled as pro/con premises with little loss. (Wachsmuth et al., 2017f)

Outline: Conclusion

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

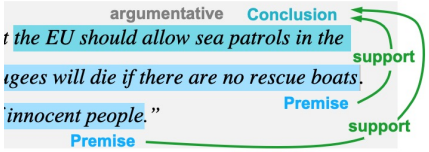
VII. Applications of computational argumentation

VIII. Conclusion

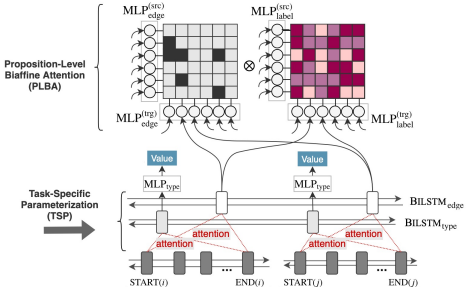
a) Introduction
b) Unit segmentation
c) Unit type classification
d) Relation identification
e) Conclusion

Conclusion

- **Argument mining**
  - Computational identification of argumentative structure
  - May be based on different argument models
  - Segmenting units, classifying types, identifying relations

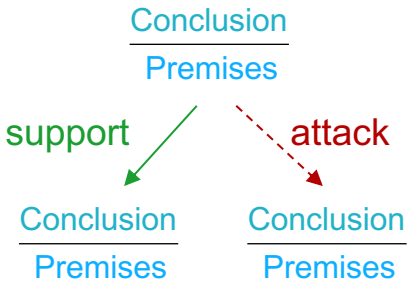


- **Selected approaches to argument mining**
  - Unit segmentation using Bi-LSTMs
  - Unit type classification using biaffine attention
  - Relation identification using MSTs, LDA, and prompting



(Moro et al., 2020)

- **Discussion of argument mining**
  - May work pretty reliable within narrow, explicit genres
  - Hard on subtle argumentation and across genres
  - Simple argument models may enable more robustness



References

- **Ajjour et al. (2017)**. Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. Unit Segmentation of Argumentative Texts. In Proceedings of the Fourth Workshop on Argument Mining, pages 118–128, 2017.
- **Al-Khatib et al. (2016).** Khalid Al Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. A News Editorial Corpus for Mining Argumentation Strategies. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3433–3443, 2016.
- **Al-Khatib et al. (2017).** Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, and Benno Stein. Patterns of Argumentation Strategies across Topics. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1362–1368, 2017.
- **Bao et al. (2021).** Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. A Neural Transition-based Model for Argumentation Mining. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6354–6364, 2021.
- **Chu and Liu (1965).** Y. J. Chu and T. H. Liu. 1965. On the Shortest Arborescence of a Directed Graph. Science Sinica, 14:1396–1400.
- **Dozat and Manning (2017).** Timothy Dozat and Christoper D. Manning. Deep Biaffine Attention for Neural Dependency Parsing. In Proceedings of the 5th International Conference on Learning Representations, 2017.
- **Eger et al. (2017).** Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. Neural end-to-end learning for computational argumentation mining. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11–22, 2017.
- **Freeman (2011).** Argument Structure: Representation and Theory. Springer, 2011.

References

- **Gorur et al. (2025).** Deniz Gorur, Antonio Rago, and Francesca Toni. Can Large Language Models perform Relation-based Argument Mining? In Proceedings of the 31st International Conference on Computational Linguistics, pages 8518–8534, 2025.
- **Habernal and Gurevych (2015).** Exploiting Debate Portals for Semi-supervised Argumentation Mining in User-generated Web Discourse. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2127– 2137, 2015.
- **Habernal and Gurevych (2017).** Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. Computational Linguistics, 43(1), pages 125–179, 2017.
- **Jiang et al. (2024).** Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Flo- rian Bressand, Gianna Lengyel, Guillaume Bour, Guil- laume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie- Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.
- **Jurafsky and Martin (2024).** Daniel Jurafsky and James H. Martin (2024). Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 3rd edition draft. https://web.stanford.edu/~jurafsky/slp3/
- **Lawrence and Reed (2017).** John Lawrence and Chris Reed. Mining Argumentative Structure from Natural Language text using Automatically Generated Premise-Conclusion Topic Models. In Proceedings of the 4th Workshop on Argument Mining, pages 39–48, 2017.
- **Morio et al. (2020).** Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. Towards Better Non-Tree Argument Mining: Proposition-Level Biaffine Parsing with Task-Specific Parameterization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3259–3266, 2020.

References

- **Niculae et al. (2017).** Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument Mining with Structured SVMs and RNNs. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 985–995, 2017.
- **Rinott et al. (2015).** Ruty Rinott, Lena Dankin, Carlos Alzate Perez, M. Mitesh Khapra, Ehud Aharoni, and Noam Slonim. Show Me Your Evidence — An Automatic Method for Context Dependent Evidence Detection. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 440–450, 2015.
- **Peldszus and Stede (2015).** Andreas Peldszus and Manfred Stede. Joint Prediction in MST-style Discourse Parsing for Argumentation Mining. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 938–948, 2015.
- **Persing and Ng (2016).** Isaac Persing and Vincent Ng. End-to-End Argumentation Mining in Student Essays. In Proceedings of NAACL-HLT 2016, pages 1384–1394, 2016.
- **Spliethöver et al. (2019).** Maximilian Spliethöver, Jonas Klaff, and Hendrik Heuer. Is It Worth the Attention? A Comparative Evaluation of Attention Layers for Argument Unit Segmentation. In Proceedings of the 6th Workshop on Argument Mining, pages 74–82, 2019.
- **Stab (2017).** Christian Stab. Argumentative Writing Support by means of Natural Language Processing, Chapter 5. PhD thesis, TU Darmstadt, 2017.
- **Stede and Schneider (2018).** Manfred Stede and Jodi Schneider. Argumentation Mining. Synthesis Lectures on Human Language Technologies 40, Morgan & Claypool, 2018.
- **Toulmin (1958).** Stephen E. Toulmin. The Uses of Argument. Cambridge University Press, 1958.

References

- **Touvron et al., (2023).** Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- **Wachsmuth et al. (2017f).** Henning Wachsmuth, Giovanni Da San Martino, Dora Kiesel, and Benno Stein. The Impact of Modeling Overall Argumentation with Tree Kernels. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2369–2379, 2017.