Computational Argumentation — Part VI

Argument Generation

Henning Wachsmuth

https://ai.uni-hannover.de

Leibniz Universität Hannover

Learning goals

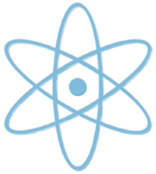
- **Concepts**
  - Selected basics of natural language generation
  - Views of core building blocks of an argument
  - Distinction of content and style in general and in text

https://commons.wikimedia.org

- **Methods**
  - Extractive and abstractive summarization of argumentative texts
  - Knowledge-based and neural techniques for generating arguments
  - Neural language models for countering arguments

https://pixabay.com

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



https://pixabay.com

- **Within this course**
  - How to reuse mined and assessed arguments in new arguments and how create fully new arguments

Outline

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

- **a) Introduction**
- b) Argument summarization
- c) Argument synthesis
- d) Counterargument synthesis
- e) Conclusion

VII. Applications of computational argumentation

VIII. Conclusion

What is argument generation?

- **Argument generation**
  - The synthesis of new argumentative units, arguments, and argumentative texts

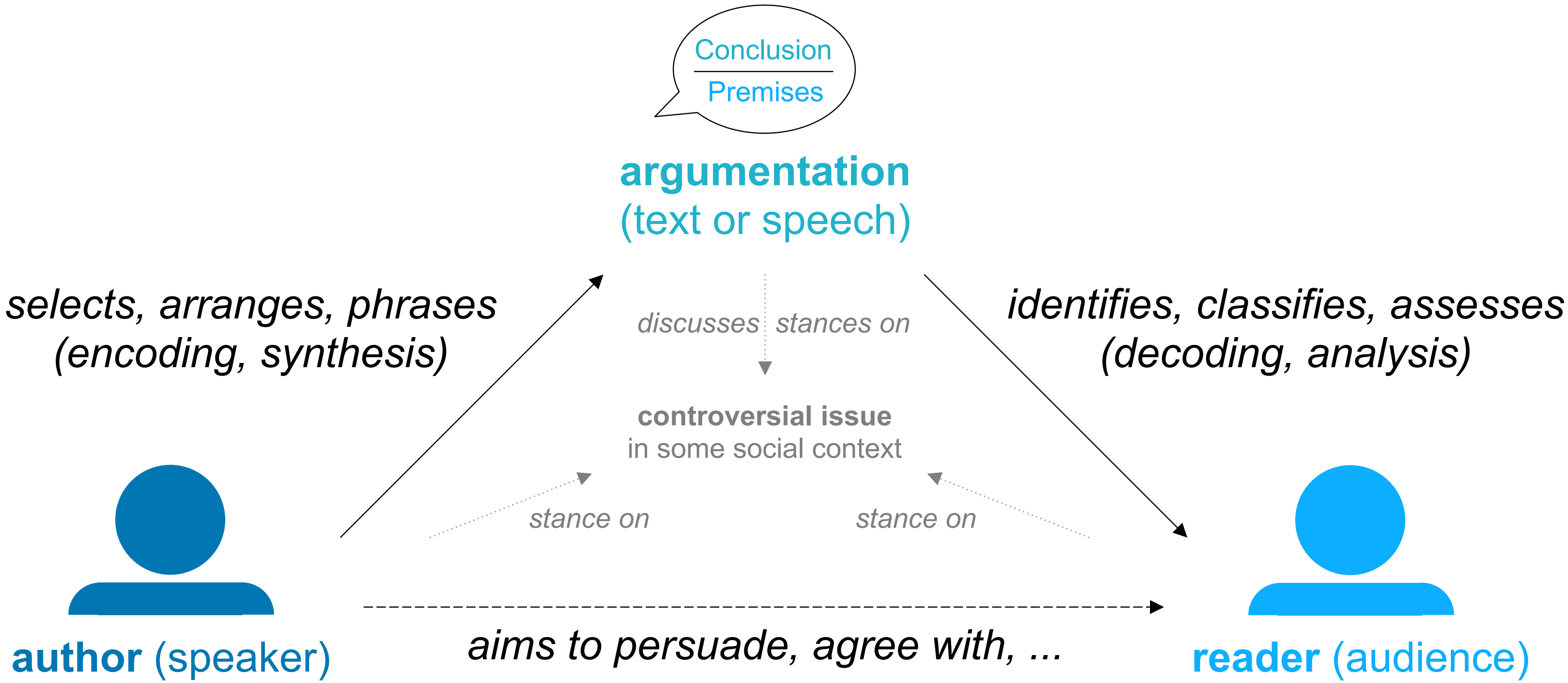
    We use *synthesis* and *generation* largely interchangeably here.

- **Argument generation tasks**
  - Writing of a summary of one or more texts
  - Encoding of knowledge in a new unit
  - Reconstruction of implicit units
  - Composition of units in an argument
  - Creation of a new argumentative text
  - Modification of existing units or arguments

    ... along with variations of these

- **Why argument generation?**
  - Technologies such as Project Debater should be able to form new arguments.
  - Computers have the potential to find new argumentative connections.

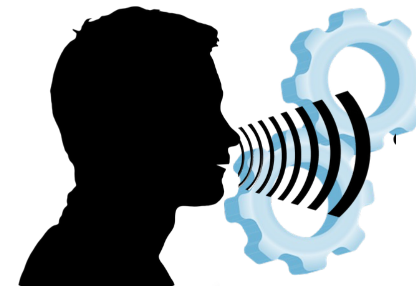
General argumentation setting (recap)



- **Generation vs. mining and assessment**
  - Argument generation refers to the encoding/synthesis side.
  - Still, mining and assessment may be required to decide what to generate.
    e.g., starting from what the opponent argued before

Natural language generation (NLG)

- **Natural language generation (NLG)**
  - Algorithms for the synthesis of natural language (text)
  - The goal is to encode structured or semi-structured information in an unstructured text



https://de.wikipedia.org
https://pixabay.com

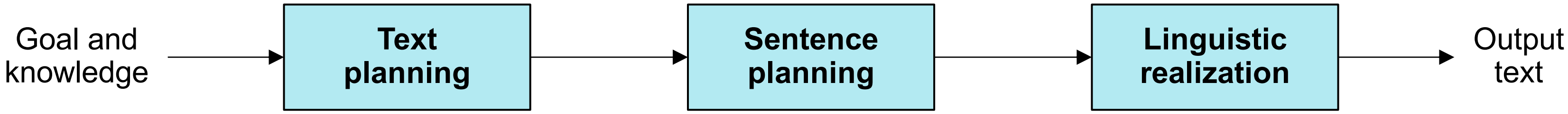
- **Two general types of NLG**
  - Data-to-text. Phrase a new text with data from a knowledge base.
  - Text-to-text. Rewrite a given text into another text.

- **What makes NLG challenging?**
  - NLG requires to *choose* and *create* a specific textual representation from many potential representations.
  - Challenges. Grammaticality, coherence, naturalness, and many more

- **Disclaimer**
  - Only a high-level introduction to selected NLG techniques is given below; more may be needed for working with NLG in general.

NLG process and techniques

- **A full NLG process** (Reiter and Dale, 1997)
  - Input. A goal of what to generate, and knowledge represented in some way
  - Output. A natural language text



- **Main steps**
  - Text planning. Select content, arrange the discourse structure of sentences
  - Sentence planning. Aggregate sentence content, make lexical choices, build referring expressions, ...
  - Linguistic realization. Orthographic, morphological, and syntactic processing

    Not all main steps (and far from all sub-steps) are always needed.

- **NLG techniques detailed below**
  - Summarization, language models, text style transfer, and more

    Often, different techniques need to be combined adequately.

Evaluation of NLG

- **How to evaluate NLG?**
  - Goal. Judge quality of generated texts.
  - Problem. There is not *the* correct output.

Ground truth. *”Ban death penalty“*

Generated text. *”We should ban the death penalty forever“*

- **Two types of NLG evaluation** (details below)
  - Automatic. All main metrics quantify word overlap between ground-truth and generated text in some way.
    Other, partly more task-specific metrics have been proposed, but are not used often (due to comparability).
  - Manual. Human annotators assess quality dimensions of generated texts.

- **Main criticisms of automatic evaluation metrics**
  - Uninterpretability. Errors are not distinguishable, not all ”errors” are wrong.
  - Unreliability. Automatic and human assessment often do not correlate.

- **Dilemma of evaluation**
  - Only manual evaluation is seen as reliable, but it costs time and money.
  - Automatic evaluation is needed to observe progress during development.

Evaluation of NLG: Automatic metrics

- **Overview of automatic metrics**
  - BLEU. Precision of *n*-gram overlap with brevity penalty
  - METEOR. F_1-score of 1-grams with word-order penalty, weighting recall 9x
  - ROUGE. Recall of *n*-gram overlap, either for a specific *n* or averaged
  - BERTScore. F_1-score derived from similarity matching of BERT embeddings

- **BiLingualEvaluation Understudy (BLEU) score**
  - Given all *n*-grams in ground-truth texts D_{gt}, and all generated *n*-grams in D_{gn}
  - Modified *n*-gram precision. Fraction of D_{gn} that matches any *n*-gram in D_{gt}, counting each *n*-gram in D_{gt} once only
  - Brevity penalty. Prevents high scores for short texts to account for recall

$$BLEU = \underbrace{\exp\Bigg(\sum_{d \in D_{gn}} \frac{1}{n} \cdot \log}_{\text{geometric mean}} \underbrace{\frac{\#ngram\ matches(d)}{\#ngrams(d)}}_{\text{modified precision for generated text } d} \Bigg) \cdot \underbrace{\exp\Bigg(\min\{1 - \frac{\#words(D_{gt})}{\#words(D_{gn})}, 0\} \Bigg)}_{\text{brevity penalty}}$$

  - This value is averaged over all considered *n*.
    Usually, *n*≤2 or *n*≤4 is used, and case sensitivity is ignored. BLEU scores are in [0,1], sometimes multiplied by 100.

Evaluation of NLG: Manual evaluation

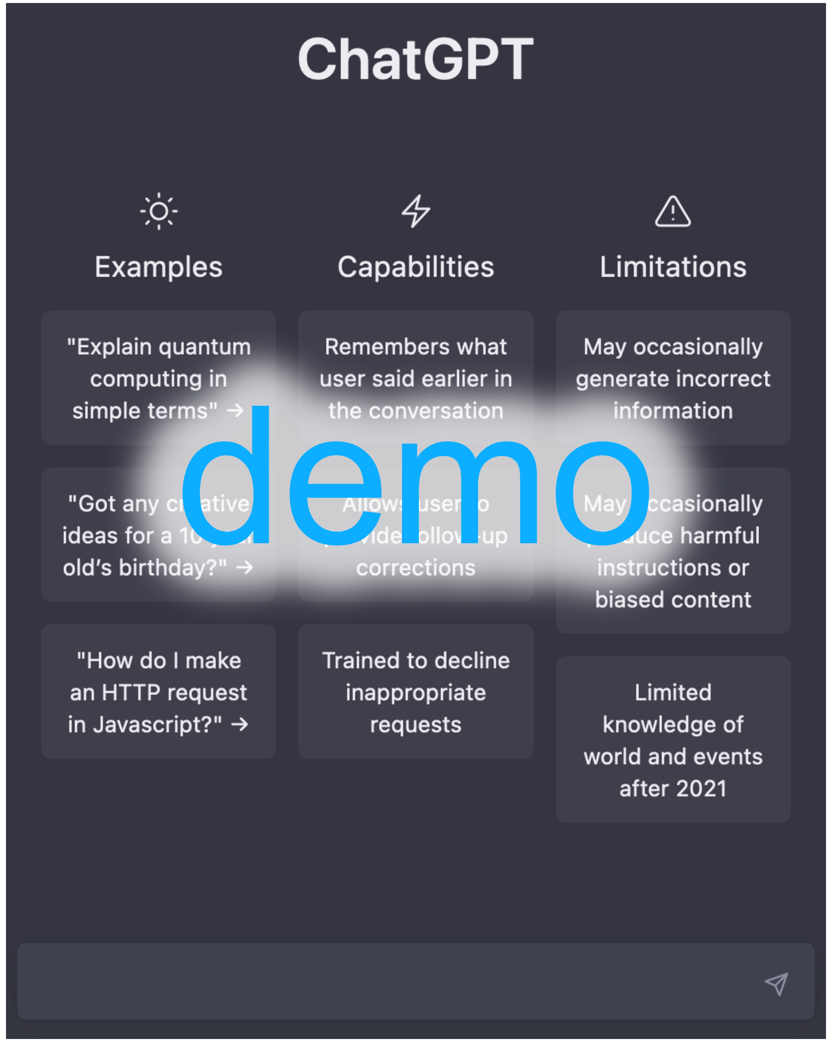
- **Manual evaluation**
  - Multiple human annotators assess the quality of a sample of generated texts.

- **Assessment**
  - Absolute scores on a Likert scale (say, 1–5) or relative ranking of candidates
  - The mean or majority judgment of annotators is used for evaluation.
  - As for corpora, inter-annotator agreement can be computed to assess reliability.

- **Quality dimensions**
  - What dimensions to be assessed, is to some extent task-specific.
  - Some dimensions are very common according to a literature survey. (van der Lee et al., 2019)

Quality dimension	#
Fluency	13
Naturalness	8
Quality	5
Meaning preservation	5
Relevance	5
Grammaticality	5
Overall quality	4
Readability	4
Clarity	3
Manipulation check	3
Informativeness	3
Correctness	3
… others with count ≤ 2	35

NLG Demo: Neural language model



https://chat.openai.com

Next section: Argument summarization

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

a) Introduction

b) Argument summarization

c) Argument synthesis

d) Counterargument synthesis

e) Conclusion

VII. Applications of computational argumentation

VIII. Conclusion

What is argument summarization?

- **Argument summarization**
  - The generation of a summary from one or more argumentative texts
  - Input. An argumentative text or a set of texts
  - Output. A summary in terms of a short text, a set of key points, or similar

Climate Change is causing the Earth to warm up measurably, and there are already signs of disaster. I argue that this is happening because there are scientific facts to prove it. Out of 918 peer-reviewed scientific papers on this subject, 0% disagreed that climate change is happening, but in newspaper articles, 53% were unsure. This proves that climate change is happening, but scientists are having trouble conveying the information and other data to the people of the world.

→ *There is no doubt that climate change is causing global warming. In a survey of 918 scientific papers, no one disagreed with this.*

Argument summarization: Example

- **How easy is summarizing arguments for humans?**
  - What would you see as the gist of the following argument pro abortion?

The Supreme Court decided that states can't outlaw abortion because Prohibiting abortion is a violation of the 14th Amendment, according to the Court, and the constitution. Outlawing abortion is taking away a human right given to women. In reality, a fetus is just a bunch of cells. It has not fully developed any vital organs like lungs. This means that an abortion is not murder, it is just killing of cells in the wound. If the child has no organs developed that would be vital for the baby to survive outside the wound, than having an abortion is not murder.

- **What makes argument summarization challenging?**
  - Argumentative texts may combine multiple claims and reasons.
  - What is most important, may be seen subjectively.
  - Unlike here, a good summary may often require rephrasing.

  ... among other challenges

Background: NLG via summarization

- **What is a summary?**
  - A short(er) text, derived from one or more long(er) texts, that presents the information important in a given context in a coherent fashion

- **Summarization**
  - The computational generation of a summary of one or more texts
  - Techniques include clustering, graph analyses, neural text generation, …

- **Extractive vs. abstractive summarization**
  - Extractive. Create summary by reusing portions of text (with no/few changes).
  - Abstractive. Reformulate core content by using new words or paraphrases.

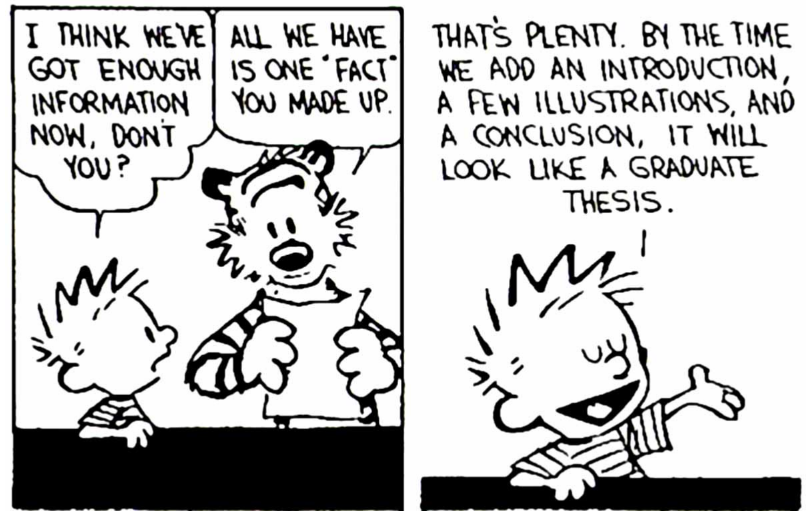
  Both are seen as generation tasks, because the output is a new text.

- **Single vs. multi-document summarization**
  - Single. Summarize the information from a single text.
  - Multi. Summarize the information from several somehow related texts.

  While the conceptual difference seems small, very different techniques are used usually.

Overview of argument summarization

- **How to model argument summarization computationally?**
  - Extractive. Identify most important units (or similar) and return them.
  - Abstractive. Reformulate the gist of the arguments in new words or paraphrases.
  - Single vs. multi. Whether the input is one argumentative text, a whole debate, or similar



- **Selected approaches to argument summarization**
  - Multi-argument keyphrase clustering for online debates (Egan et al., 2016)
  - Abstractive summarization of texts using neural models (Wang and Ling, 2016)
  - Learning-based mapping of arguments to key points (Bar-Haim et al., 2020)
  - Extractive summarization of arguments with graph methods (Alshomary et al., 2020b)
  - Knowledge-augmented generation of informative conclusions (Syed et al., 2021)

Abstractive summarization of texts (Wang and Ling, 2016)

- **Task**
  - Given several reasons on an issue, summarize them into one claim.
- **Approach**
  - Neural sequence-to-sequence model that reads reasons and writes a claim
  - First, a subset of the reasons is sampled by scoring their value for a summary.
  - Then, an attention-based LSTM learns long-term dependencies.
- **Data**
  - 676 debates with 2259 claims and 17,359 reasons from idebate.org.
- **Results**
  - BLEU 25.8 (Best extractive baseline 15.1)

    Not better in terms of METEOR and ROUGE

Issue: *This House would detain terror suspects without trial.*

(1) Governments must have powers to protect their citizens against threats to the life of the nation. (2) Everyone would recognise that rules that are applied in peacetime may not be appropriate during wartime.

Human. *Governments must have powers to protect citizens from harm.*

Approach. *Governments have the obligation to protect citizens from harmful substances.*

Extractive summarization of arguments (Alshomary et al., 2020b)

- **Task**
  - Given an argumentative text, generate a two-sentence *snippet* that best represents the gist of the argumentation.

> *The Supreme Court decided that states can't outlaw abortion because Prohibiting abortion is a violation of the 14th Amendment, according to the Court, and the constitution. Outlawing abortion is taking away a human right given to women. In reality, a fetus is just a bunch of cells. It has not fully developed any vital organs like lungs. This means that an abortion is not murder, it is just killing of cells in the wound. If the child has no organs developed that would be vital for the baby to survive outside the wound, than having an abortion is not murder.*

→

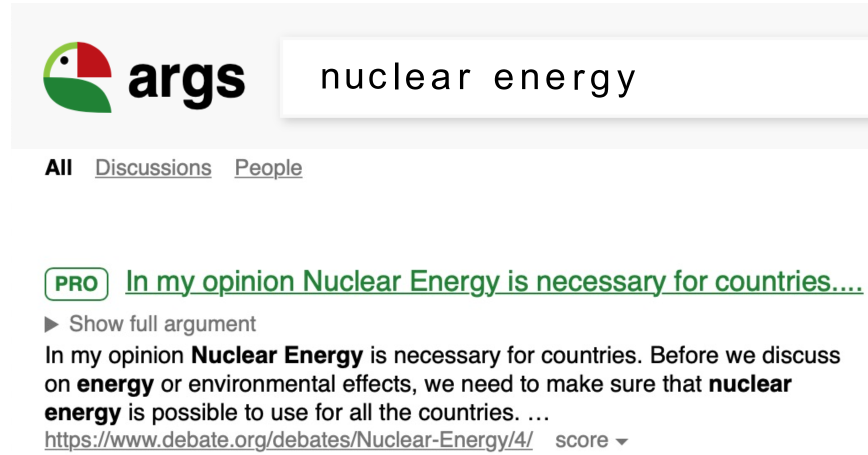
> *In reality, a fetus is just a bunch of cells. This means that an abortion is not murder, it is just killing of cells in the wound.*

- **Research question**
  - How important are the context and argumentativeness of a sentence?

- **Approach in a nutshell**
  - Compute a representativeness score of each sentence from its centrality in its context and its argumentativeness.
  - Return the two sentences with highest score in their original ordering.

Extractive summarization of arguments: Snippets

- **Snippet**
  - A short text that helps assess the relevance of a search result
  - In general web search, a snippet usually shows a content excerpt containing the query terms.

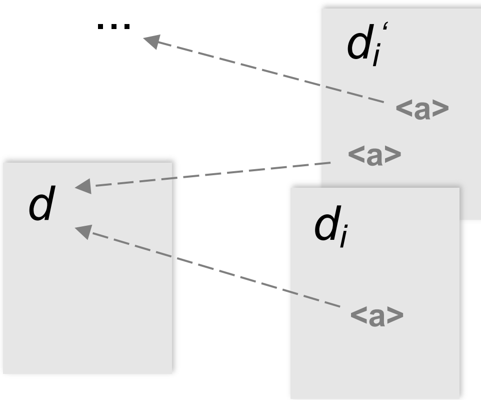


- **Snippets in argument search**
  - Snippets are key to get an efficient overview of search results.
  - This is of special importance in argument search, where it is often not enough to obtain only one relevant result.
  - Standard snippets may be not enough for arguments. (as in the example above)

- **What is a good argument snippet?**
  - Hypothesis. A short text representing the gist of an argument, in terms of the main claim and main reason supporting the claim
  - The approach presented here generates *query-independent* snippets.

Extractive summarization of arguments: Approach

- **PageRank** (recap)
  - An unsupervised method to recursively assess the objective importance of a web page
  - Main idea. A page is more important the more other important pages link to it.



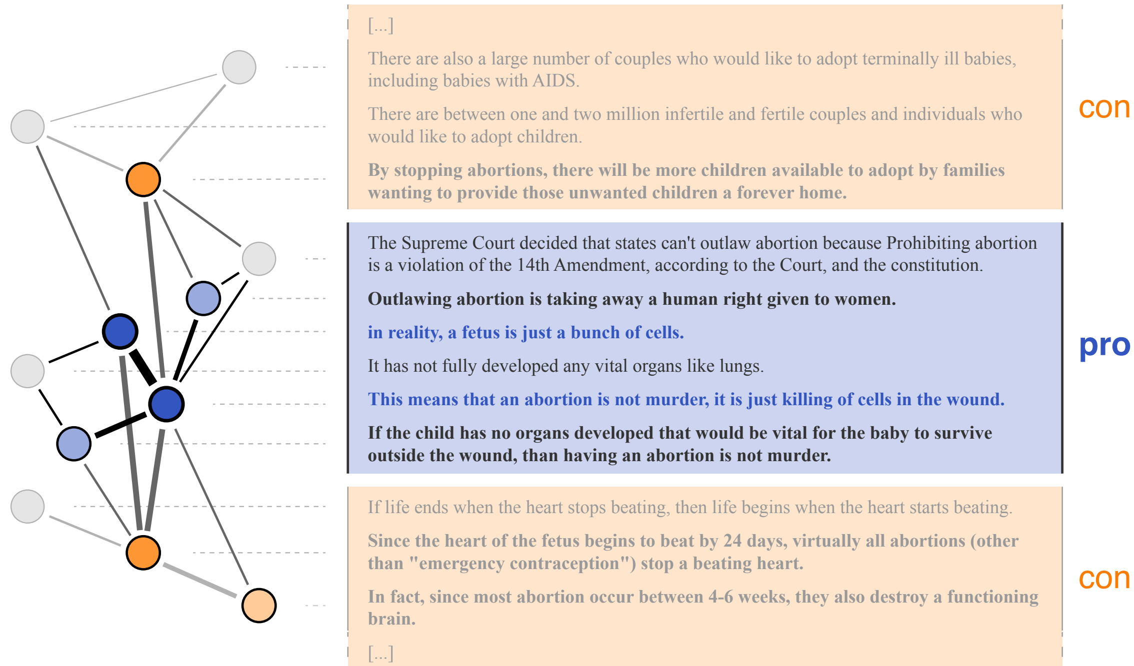
- **LexRank** (Erkan and Radev, 2004)
  - Adaptation of PageRank to assess the *centrality* of a sentence in a text
  - Main idea. A sentence is more important the more similar it is to other important sentences in the same text.

- **LexRank for extractive summarization**
  - Compute LexRank score for all sentences in the context of an argument.
  - Bias the score to sentences that are *argumentative*.

$$P(s_i) = (1-\alpha) \cdot \sum_{s_j \neq s_i} \underbrace{\frac{sim(s_i, s_j)}{\sum_{s_k \neq s_j} sim(s_j, s_k)}}_{\text{Centrality as "exclusive" sentence similarity}} P(s_j) + \alpha \cdot \underbrace{\frac{arg(s_i)}{\sum_{s_k} arg(s_k)}}_{\text{Bias to argumentative sentences (normalized)}}$$

Extractive summarization of arguments: Realization

- **How to model context?**
  - For debates, the other arguments there serve as suitable context.
  - Otherwise, arguments could be clustered into contexts.

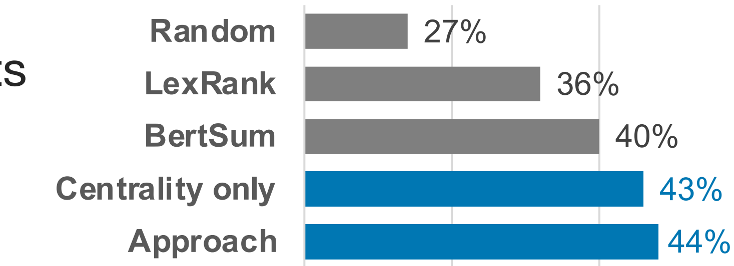


- **How to compute centrality and argumentativeness?**
  - Centrality. Cosine similarity between the sentences‘ embeddings
    Sentence embeddings generalize the idea of word embeddings to sentences.
  - Argumentativeness. Frequency of words from a discourse lexicon
    Argument mining performed worse in experiments, possibly due to heterogeneous input.

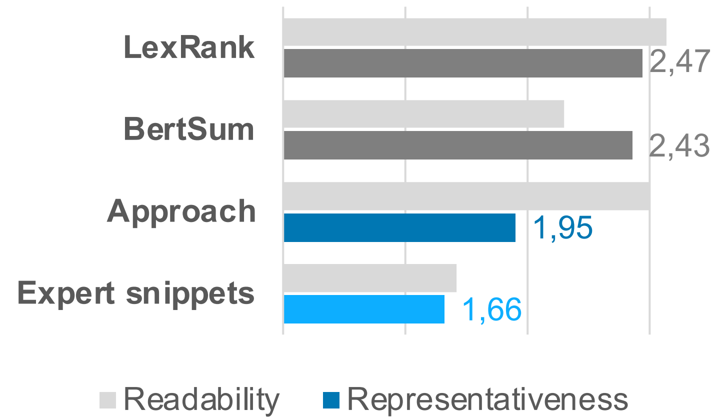
- **Notice**
  - These are realization details that could be replaced.

Extractive summarization of arguments: Results

- **Evaluation**
  - Data. Expert snippets for 50 args.me results
  - Automatic. Accuracy of snippet generation
  - Manual. Mean rank of representativeness and readability (from 3 annotators)

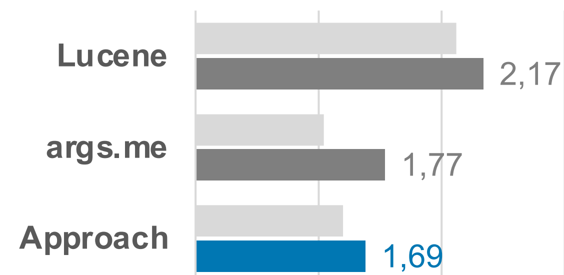


- **Extractive summarization baselines**
  - Random. Selecting any 2 sentences
  - LexRank. Simple PageRank for sentences
  - BertSum. Neural extractive summarization
  - Expert snippets. Ground truth



- **Existing snippet generation baselines**
  - Lucene. Query-dependent snippet generation
  - args.me. Using the beginning of arguments

(all snippets cut after 225 characters to mimic real application)



Extractive summarization of arguments: Example

- **Argument returned to query ”climate change“**

Climate Change is causing the Earth to warm up measurably, and there are already signs of disaster. I argue that this is happening because there are scientific facts to prove it. Out of 918 peer-reviewed scientific papers on this subject, 0% disagreed that climate change is happening, but in newspaper articles, 53% were unsure. This proves that climate change is happening, but scientists are having trouble conveying the information and other data to the people of the world.

- **Which snippet best represents the argument’s gist?**

#1. *Climate Change is causing the Earth to warm up measurably, and there are already signs of disaster… I argue that this is happening because there are scientific facts to prove it…*

args.me

#2. *Out of 918 peer-reviewed scientific papers on this subject, 0% disagreed that climate change is happening, but in newspaper articles, 53% were unsure… This proves that climate change is happening, …*

approach

#3. *Climate Change is causing the Earth to warm up measurably, and there are already signs of disaster … reviewed scientific papers on this subject, 0% disagreed that climate …*

Lucene

Argument summarization: Discussion

- **Complexity of argument summarization**
  - Summarization is a hard task in general, but has seen notable progress with recent neural architectures.
  - Abstractive summarization is more challenging, but more human-like.
  - As usual, the more narrow the domain of texts, the better it may work.

- **Good argument summaries**
  - An argument summary should represent the main reasoning well.
  - How much subjectiveness should be kept, depends on the application.
  - Research on how to best summarize argumentation is still limited.
    The work of Alshomary et al. (2020b) was the first to explicitly raise this question.

- **Why argument summarization?**
  - Not only in argument search, short argument summaries are needed.
  - Getting an overview of different or longer arguments is important in many applications of computational argumentation.
    Rationale behind: We cannot always consume all information out there.

Next section: Argument creation

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

a) Introduction
b) Argument summarization
c) Argument synthesis
d) Counterargument synthesis
e) Conclusion

VII. Applications of computational argumentation

VIII. Conclusion

What is argument synthesis?

- **Argument synthesis**
  - The creation of argumentative units, arguments, or full argumentative texts
    Various possible task definitions

- **Example: Claim synthesis**
  - Input. An issue, along with knowledge on the issue represented in some way
  - Output. A unit conveying a stance towards the issue

Rescue boats → *"Having **rescue boats** makes even more **people die** trying."*

- **Example: Argumentative text synthesis**
  - Input. An issue and stance, along with knowledge represented in some way
  - Output. A text arguing towards the given stance on the given issue

Pro *Rescue boats* → *" 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. While having rescue boats may make even more people die trying, nothing justifies to endanger the life of innocent people. Got it?"*

Argument synthesis: Examples and challenges

- **How easy is argument synthesis for humans?**
  - Given the following pool of concepts and predicates, phrase reasonable units.

    Examples adapted from Bilu and Slonim (2016)

great anarchy *a global language* *contribute to stability* *is a source of conflict*

democratization *lead to great exhaustion*

Democratization contributes to stability.

  - Given the following claim, phrase a meaningful reason for it.

A university degree is important for your career.

Employers look at what degree you have first.

- **Challenges of argument synthesis**
  - Knowledge bases might not contain suitable concepts for everything.
  - Linguistic adaptations of grammar may be necessary.
  - Connections between different concepts build on world knowledge.
  - Content, reasoning, and stance all need to be encoded properly.

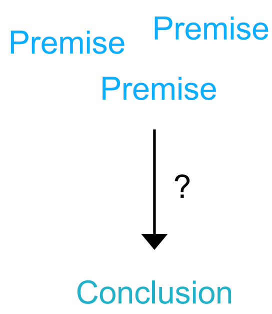
Overview of argument synthesis

- **How to model argument synthesis computationally?**
  - Approaches vary notably, due to differences in task definitions.
  - Composition. Fill templates with concepts from knowledge bases or other text
  - Language modeling. Generate free text based on trigger concepts.
  - Controlled generation. Generate free text that fulfills specified constraints

- **Selected argument synthesis approaches**
  - Discourse planning for argumentative texts (Zukerman et al., 2000; Carenini and Moore, 2006)
  - Knowledge-based scoring for argument composition (Reisert et al., 2015; Sato et al., 2015)
  - Predicate recycling for composing new claims (Bilu and Slonim, 2016; 2019)
  - Language modeling for rhetorical argument composition (El Baff et al., 2019)
  - Neural target inference in conclusion generation (Alshomary et al., 2020a)
  - Neural knowledge encoding in argument generation (Al-Khatib et al., 2021)
  - Transformer-based generation of conclusions for assessment (Gurcke et al., 2021)
  - Conditioned neural generation of claims with beliefs (Alshomary et al., 2021a)

Target inference in conclusion generation (Alshomary et al., 2020a)

- **Task**
  - Given the premises of an argument, infer the target of its conclusion.
  - Motivation. Humans often leave parts of arguments implicit.
    Particularly, conclusions often left out (Habernal and Gurevych, 2015)

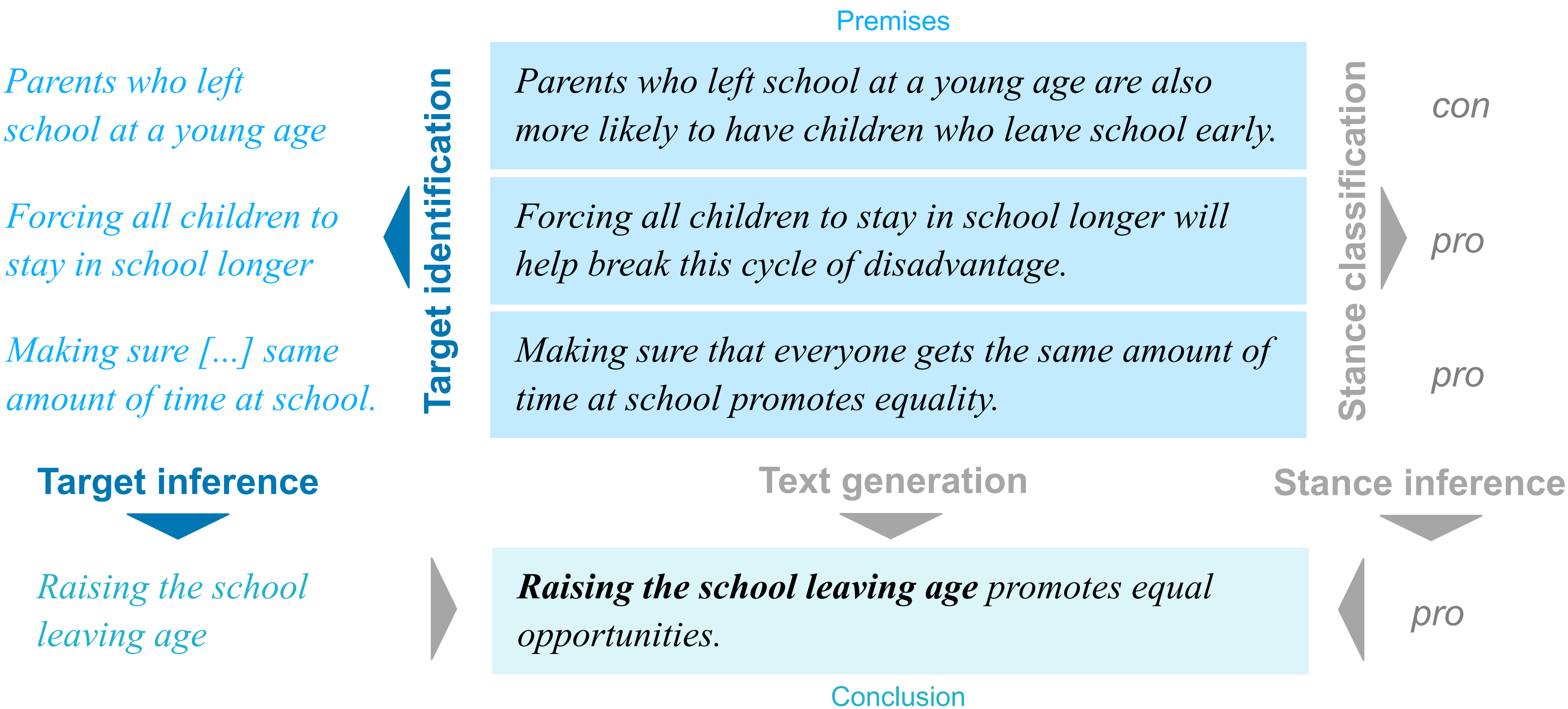


- **Hypothesis**
  - The conclusion target is related to the targets of the premises.

- **Data**
  - iDebate. 2259 arguments (Wang and Ling, 2016)
  - Essays-c. 2020 premise-conclusion arguments (Stab, 2017)
  - Essays-t. 402 conclusion-thesis arguments (Stab, 2017)
    Each split into training, validation, and test set.

- **Approach in a nutshell** (two complementary sub-approaches)
  - Either, rank identified premise targets by their representativeness.
  - Or, match generated target embedding with target knowledge base.

Target inference: Overall idea + Target identification



- **Target identification**
  - Neural tagger trained on existing data (Bar-Haim et al., 2017; Akbik et al., 2018)

Forcing all children to stay in school longer will help break this cycle of disadvantage .
B I I I I I I I O O O O O O O O

Target inference: Approach a_1

- **Inference hypothesis H_1**
  - One of the premise targets represents an adequate conclusion target

- **Approach a_1: Premise target ranking**
  - Model. Prediction of a representativeness score for each candidate target
    Trained on Jaccard similarity of ground-truth premise and conclusion targets (Wang and Ling, 2016)
  - Features. Length, position, sentiment, and similarity to other candidates
  - Inference. Pick most representative premise target

Parents who left school at a young age	0.3		
Forcing all children to stay in school longer	0.7	~	*Raising the school leaving age*
Making sure [...] same amount of time at school.	0.2		

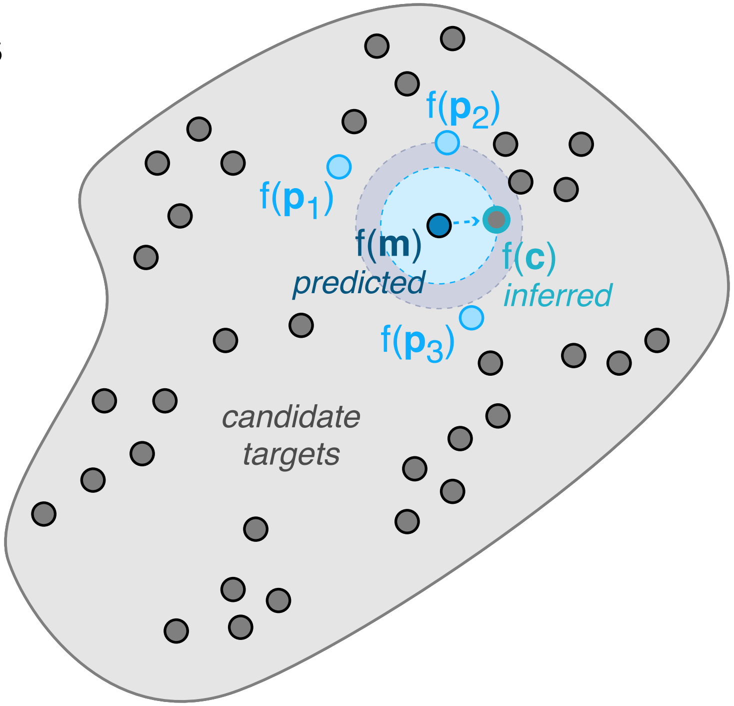
- **Implication**
  - A target that is not given in the premises can never be predicted.

Target inference: Approach a_2

- **Inference hypothesis H_2**
  - The premise targets are semantically related to adequate conclusion targets

- **Approach a_2: Embedding learning**
  - Learn to map premise target embeddings to conclusion target embedding
    Details on next slides.
  - Embed candidate targets from some knowledge base
  - Pick candidate target whose embedding is closest to premise targets

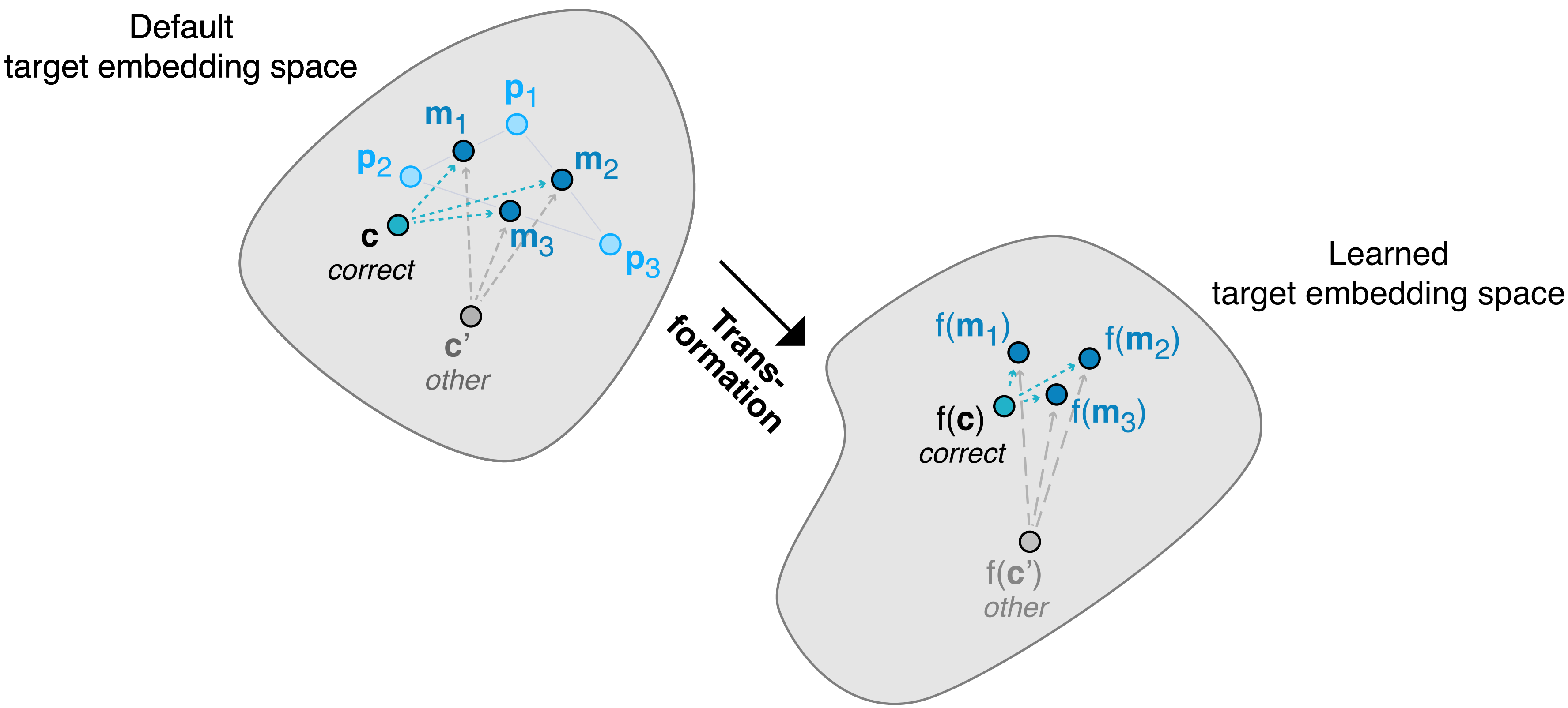
Learned target embedding space



- **Implications**
  - Guarantees to obtain a meaningful target
  - Depends on quality of target knowledge base
    Below, we use targets identified in training arguments

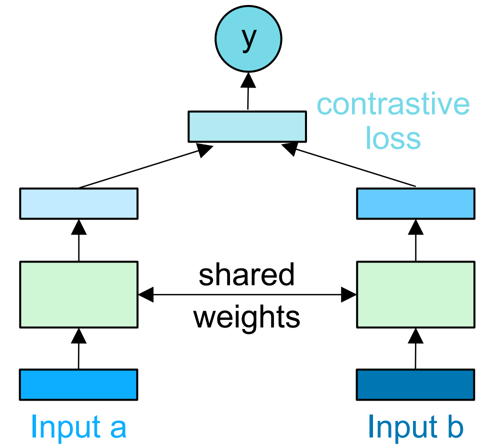
Target inference: Approach a_2 – Embedding learning

- **How to map target embeddings**
  - Compute means $\mathbf{m}_1, \ldots, \mathbf{m}_l$ of premise target embeddings $\mathbf{p}_1, \ldots, \mathbf{p}_k$.
  - Learn model f that makes $\mathbf{m}_i$ more similar to correct conclusion target $f(\mathbf{c})$ and less similar to other targets $f(\mathbf{c}')$.

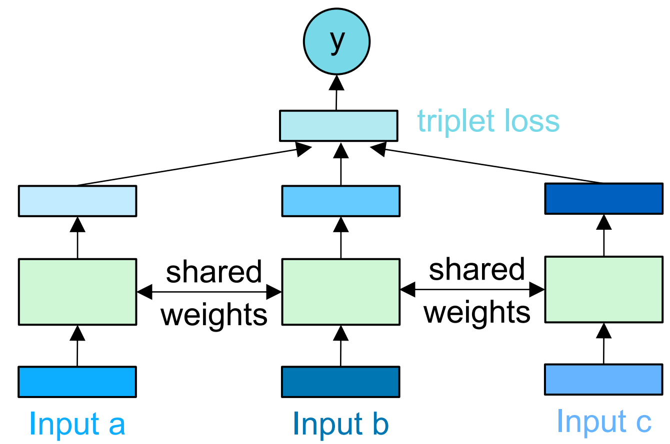


Background: Siamese and triplet neural networks

- **Siamese neural network (SNN)**
  - Two networks sharing the same weights to transform two inputs *a* and *b* into two outputs
    Thereby, the inputs are mapped to a learned embedding space.
  - The difference of outputs is quantified as a distance *d*.
  - Contrastive loss function. Minimize *d* between similar inputs, and maximize *d* for dissimilar inputs.

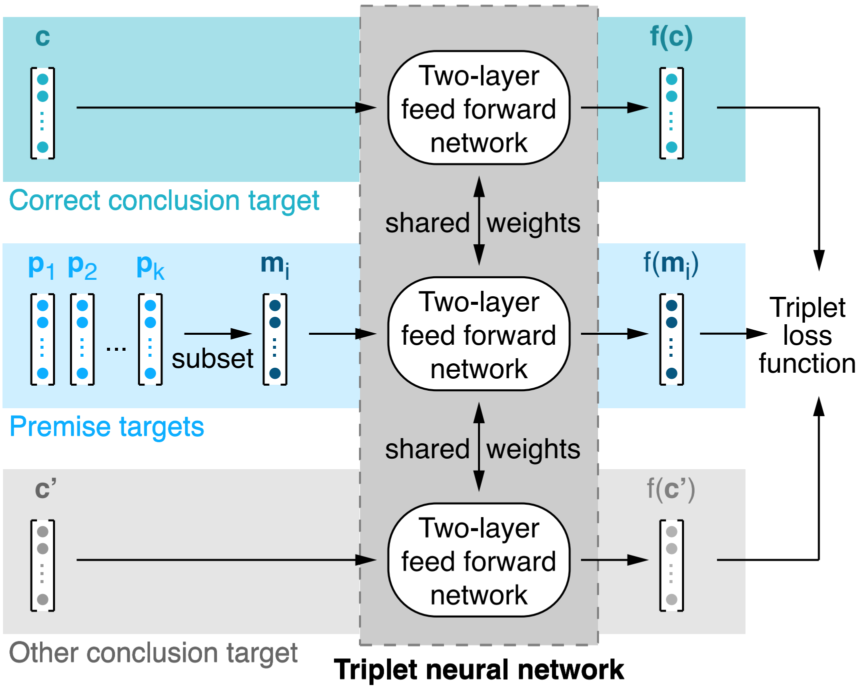


- **Triplet neural network (TNN)**
  - A TNN follows a similar idea to an SNN for three inputs *a, b, c*.
  - The input *b* is used to define distance *d*.
  - Triplet loss function. Minimize *d* between *a* and *b*, and maximize *d* for *b* and *c.*



Target inference: Approach a_2 – TNN optimization

- **How to learn the mapping**
  - Train TNN on targets from complete arguments



  - Optimize loss function based on distance to correct and to other target:

$$\mathcal{L} = \max\left\{\underbrace{d(f(\mathbf{m}_i), f(\mathbf{c}))}_{\text{Distance to correct target}} - \underbrace{d(f(\mathbf{m}_i), f(\mathbf{c}'))}_{\text{to wrong target}} + \underbrace{d_{max}}_{\text{considered maximum (hyperparamater)}},\ 0\right\}$$

Target inference in conclusion generation: Results

- **Baselines and hybrid approach**
  - Seq2Seq*. Summarize premises, tuned to their targets (Wang and Ling, 2016)
  - Premise target (random). Pick one premise target randomly.
  - Embedding (mean). Pick candidate that most resembles premise targets.
  - Hybrid approach. a_2 if inferred target overlaps with any premise, otherwise a_1
- **Evaluation**
  - Automatic. BLEU score for 1- and 2-grams on each dataset
  - Manual. Percentage of fully/somewhat adequate targets (only on iDebate)

Approach	iDebate	Essays-c	Essays-t	Fully	Somewhat	Not
Seq2Seq*	4.4	–	–	5%	18%	**76%**
Premise target (random)	3.9	2.2	8.8	–	–	–
Embedding (mean)	7.2	**8.3**	15.3	–	–	–
Premise target ranking	9.7	4.1	17.3	**56%**	33%	11%
Embedding learning	9.2	**8.3**	**27.9**	50%	28%	22%
Hybrid approach	**10.0**	8.2	**27.9**	55%	**34%**	11%

Target inference in conclusion generation: Examples

- **Input: A set of premise targets**

Example 1	Example 2	Example 3
how to use the mobile phone	*Relocating to the best universities*	*Saving the use of that kinds of languages*
Phones	*Improving the pool of students*	*in this case*
Having a mobile phone	*Online courses*	*to be respected and preserved*
the internet phones	*Stanford University's online course on Artificial Intelligence*	*language*

- **Output: One conclusion target**

	Example 1	Example 2	Example 3
ground truth	*Mobile phones*	*Online courses*	*the government*
ranking	*Phones*	*Online courses*	*language*
inference	*Mobile phones*	*distance-learning*	*language acquisition*

Generation of conclusions for assessment (Gurcke et al., 2021)

- **Assessment task**
  - (Local) Sufficiency. An argument‘s premises make it rationally worthy to draw the conclusion. (Johnson and Blair, 2006)
  - Given an argument, decide whether it is sufficient or not.

Conclusion

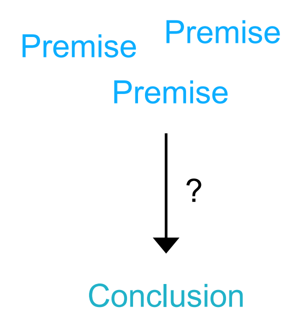
Last, ***we should develop at least one personal hobby, not to show off, but express our emotion when we feel depressed or pressured****. Playing musical instrument is a good way, I can play guitar. When I meet difficulties in studies, I take my guitar and play the song Green Sleeves. It makes me feel better and gives me confidence.*

Premises

→ **Insufficient**

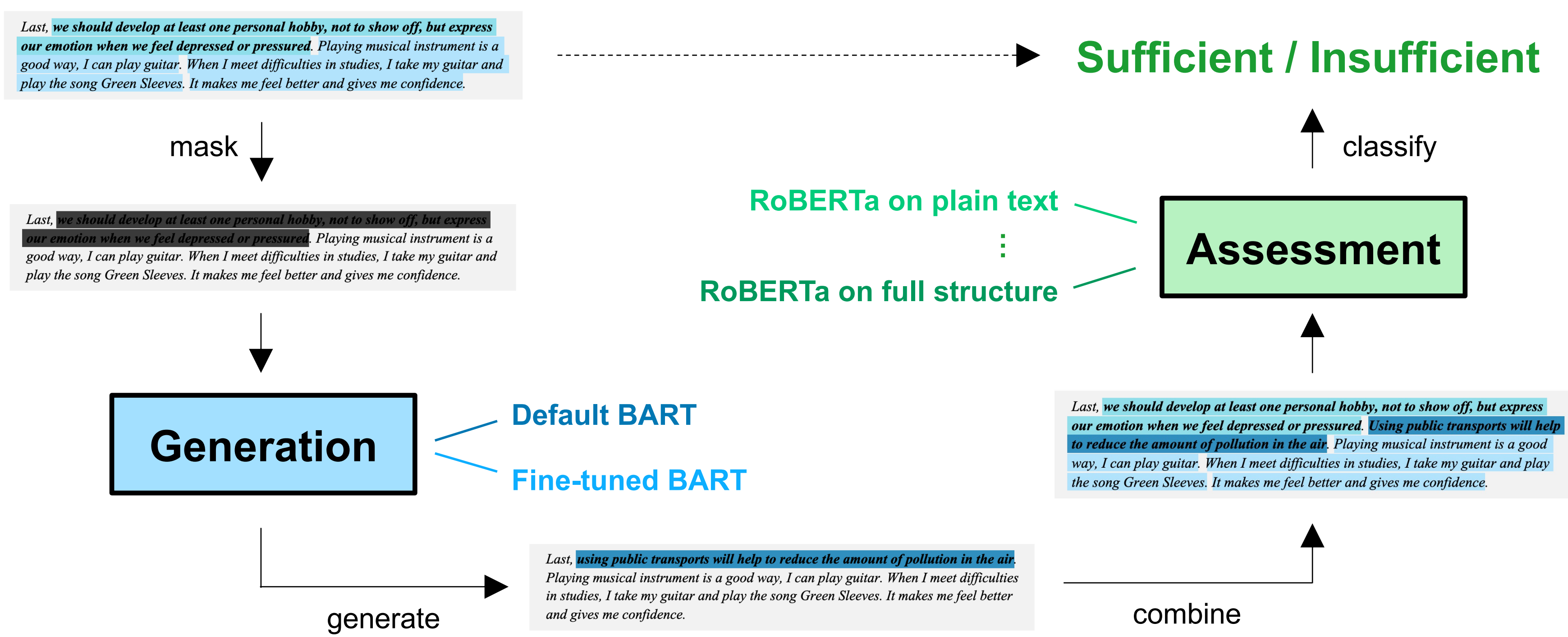
- **Research question**
  - How is local sufficiency reflected in language?

- **Generation task**
  - Hypothesis. Only for sufficient arguments, the conclusion can be *inferred* from their premises.
  - Given an argument‘s premises, generate the conclusion.



Generation of conclusions for assessment: Approach

- **Approach in a nutshell**
  - Generation. Infer a(nother) conclusion from the argument‘s premises.
  - Assessment. Classify local sufficiency based on the full argument and the inferred conclusion



Background: NLG via language models

- **Language model**
  - A probability distribution over a sequence of words
    A language model assigns a probability $P(w_1, ..., w_m)$ to each sequence of words $w_1, ..., w_m$ for any length m.
  - *n*-gram model. Approximates the probability of *m* words for some *n* as:

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i | w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

- **Language models in NLG**
  - Given an *n*-gram, the most likely words following it can directly be computed.
  - Example. 2-gram model

$P(fish\|fish) = 0.2$	$P(fish\|people) = 0.6$
$P(people\|fish) = 0.8$	$P(people\|people) = 0.4$

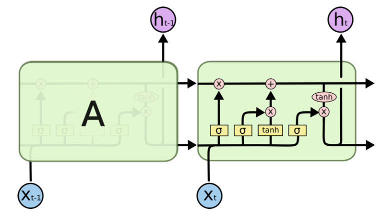
Input. *"fish"*

"... people fish"	0.48
"... fish people"	0.16

- **How to build a language model?**
  - Statistical. Compute probabilities from word sequences in a corpus.
  - Neural. Represent words by embeddings and derive probabilities accordingly.
    The higher *n*, the more data is needed for reliable probablities. Neural models are built on up to billions of texts.

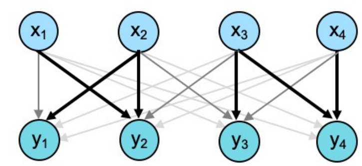
Background: Transformer neural networks

- **LSTM: Recap and problem**
  - RNN with memory to model long-term dependencies
  - Training is slow due to sequential input processing.
  - Long-term memory is still limited by hidden state size.

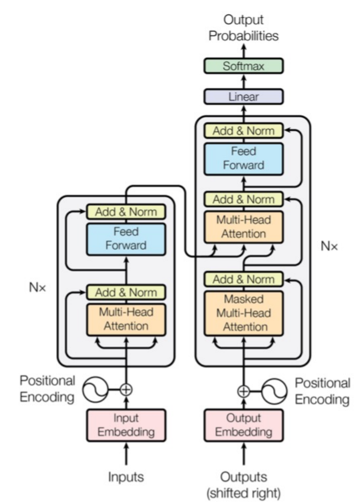


https://colah.github.io

- **Attention as a solution?**
  - Retain hidden states to model input-output dependencies
  - Self-attention to model interdependencies of inputs



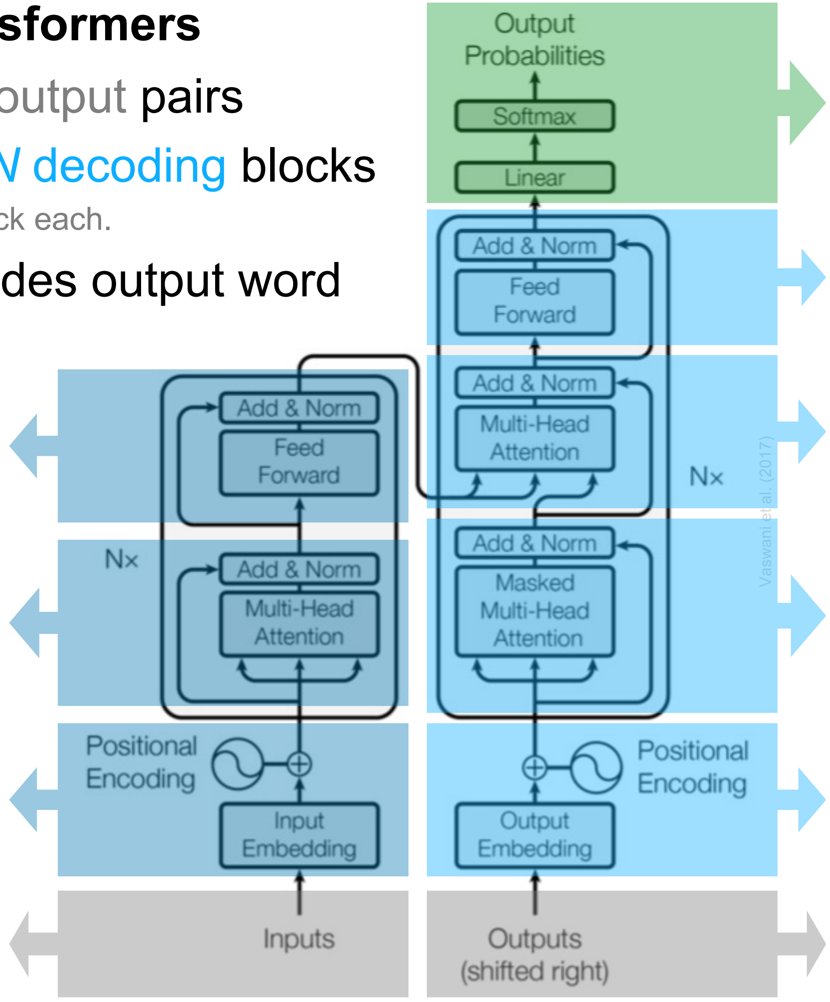
- **Transformer** (Vaswani et al., 2017)
  - A network architecture for sequence-to-sequence generation
    Can be seen as the current state of the art technique in NLP
  - Idea: Make inputs independent while modeling their context.
  - Transformers are based entirely on self-attention.
  - Faster training due to parallel processing of sequential input
  - Largely solves the modeling of long-term dependencies



Vaswani et al. (2017)

Background: Encoding and decoding of transformers

- **Architecture of transformers**
  - Trained on input/output pairs
  - *N* encoding and *N* decoding blocks
    Illustration shows one block each.
  - Output head decides output word



Map to probabilities of all possible output words (separately for each word)

Transform vector into accepted input length

Model relations between input and output words

Compute weighted average of multiple self-attention vectors, only based on previous words

Add positional information to word embedding

Sequence of output words (separately embedded)

Transform vector into accepted input length

Compute weighted average of multiple self-attention vectors

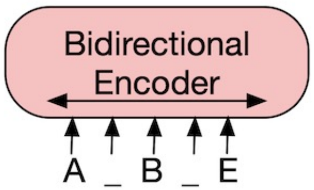
Add positional information to word embedding

Sequence of input words (separately embedded)

Death penalty kills innocent people. `<start>` Death penalty should be banned.

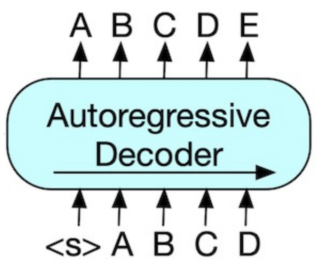
Background: Common transformer variations

- **BERT** (Devlin et al., 2019)
  - Transformer *encoder* for contextualized word embeddings
  - Pretrained for missing word prediction on 3.3B words
  - Can be fine-tuned for various tasks via added output heads



Lewis et al. (2020)

- **RoBERTa** (Liu et al., 2019)
  - More robustly trained version of BERT, often used for classification
  - Pretraining and hyperparameter optimization on ~10x as many words
- **GPT** (Radford et al., 2018)
  - Transformer *decoder* for sequence-to-sequence generation
  - Auto-regressive: Next output depends on previous output
  - GPT-3 pretrained on 470x as many words (let alone GPT-4)



Lewis et al. (2020)

- **BART** (Lewis et al., 2019)
  - Combines BERT-like encoder with GPT-like decoder
  - Robustly pretrained for various tasks on similar data size as RoBERTa

Generation of conclusions for assessment: Evaluation

- **Experimental setup**
  - Data. 100 arguments (50% sufficient) from student essays (Stab and Gurevych, 2017)
  - Approaches. Ground truth, default BART, fine-tuned BART
  - Experiments. 5 humans scored 3 relatedness dimensions, scale 1–5

- **Relatedness dimensions**
  - Novelty. How different is the conclusion from the premises?
  - Likeliness. How likely is it to infer the conclusion from the premises?
  - Sufficiency. Are the premises sufficient to draw the conclusion?

- **Manual evaluation results** (mean scores, higher is better)

Approach	Novelty	Likeliness	Sufficiency
Default BART	3.34	2.76	2.87
Fine-tuned BART	3.47	2.96	2.87
Ground truth	**3.79**	**2.98**	**2.92**

Generation of conclusions for assessment: Examples

- **Sufficient argument**

*Second, ~~**public transportation helps to solve the air pollution problems**~~. Averagely, public transports use much less gasoline to carry people than private cars. It means that by using public transports, less gas exhaust is pumped to the air and people will no longer have to bear the stuffy situation on the roads, which is always full of fumes.*

public transport is more efficient than private cars

Default BART

using public transports will help to reduce the amount of pollution in the air

Fine-tuned BART

- **Insufficient argument**

*Last, ~~**we should develop at least one personal hobby, not to show off, but express our emotion when we feel depressed or pressured**~~. Playing musical instrument is a good way, I can play guitar. When I meet difficulties in studies, I take my guitar and play the song Green Sleeves. It makes me feel better and gives me confidence.*

but not least, I love music

playing musical instrument is very important to me

Generation of conclusions for assessment: Sufficiency

- **Experimental setup** replication of (Stab and Gurevych, 2017)
  - Data. 1029 arguments (66% sufficient) from 402 student essays
  - Approaches. CNN baseline, RoBERTa on various input configurations
  - Experiments. 5-fold cross-validation, 20 repetitions

- **Input configurations**
  - Plain text compared to varying subsets of annotated argument units

- **Results** (higher is better)

Approach	Input	Macro F1
RoBERTa (our approach)	Full plain text w/o structure	.876
	Premises only	.875
	Premises + generated conclusion	.878
	Premises + original conclusion	**.885**
	Premises + both conclusions	**.885**
CNN (Stab and Gurevych, 2017)	Full plain text w/o structure	.831

Argument synthesis: Discussion

- **Effective argument synthesis**
  - A grammatically correct text can be generated easily based on templates.
  - The challenge lies in the generation of coherent, relevant, and meaningful text in a given context.
  - Recent transformer models show high effectiveness on argument generation.

- **How to synthesize arguments in practice?**
  - Not one best model: how to best synthesize depends on the setting.
  - Approaches that compose existing units may be more reliable.
  - More free (neural) text generation is currently the default, also for arguments.

- **Why argument synthesis?**
  - Increase of the capabilities of debating technologies, such as Project Debater
  - Support in argumentative writing through auto-completion or similar
  - Potential creation of really new, not yet known arguments?

Next section: Argument reconstruction

I. Introduction to computational argumentation

II. Basics of natural language processing

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

a) Introduction

b) Argument summarization

c) Argument synthesis

d) Counterargument synthesis

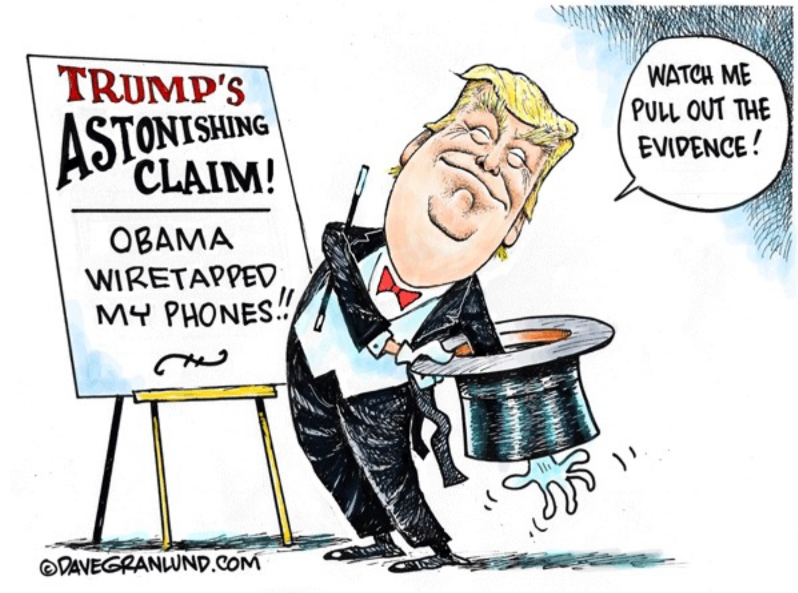
e) Conclusion

VII. Applications of computational argumentation

VIII. Conclusion

What is counterargument synthesis?

- **Counterargument synthesis**
  - The generation of a counterargument (or unit) to a given argument (or unit)
  - Input. An argument
  - Output. Another argument attacking or opposing to the argument



Argument		Counterargument
The EU should allow rescue boats in the Mediterranean Sea. Many innocent refugees will die if there are no rescue boats.	→	*Having rescue boats also may have negative effects. Even more people may die trying, believing that they may be rescued.*

Counterargument synthesis: Examples and challenges

- **How easy is counterargument synthesis for humans?**
  - Given an argument, phrase an argument undercutting its reasoning.

Abortion must be banned. It kills human life and can be hence be considered murder.

In reality, a fetus is just a bunch of cells. This means that an abortion is not murder, it is just killing of cells.

  - Given the following claim *con* Trump‘s decision, rephrase it to a *pro* claim.

Trump is making a huge mistake on Jerusalem

Trump is right in recognizing Jerusalem as Israel‘s capital

- **Challenges of counterargument synthesis**
  - Most challenges of argument synthesis also show up here.
  - Stance needs to be flipped, while clear relations of content are maintained.
  - Ultimately, the generated counter should be ”truthful“.

Overview of counterargument synthesis

- **How to model counterargument synthesis computationally?**
  - As with unit synthesis, diverse variations of the task exist.
  - Sequence-to-sequence. Given a text, rewrite it into another text.
  - Retrieve-delete-rephrase. Find, compose, and possibly adjust relevant units.
  - Language modeling. Generate free text based on trigger concepts.

- **Selected counterargument synthesis approaches**
  - Learning to rank based on joint similarity and dissimilarity (Wachsmuth et al., 2018a)
    Technically, this is not a *generation* approach.
  - Retrieval and neural generation of counters (Hua and Wang, 2018; Hua et al., 2019)
  - Neural style transfer for bias modification (Chen et al., 2018)
  - Sequence-to-sequence generation of opposing claims (Hidey and McKeown, 2019)
  - Neural generation of aspect-based counterarguments (Schiller et al., 2020)
  - Conditioned neural generation of premise attacks (Alshomary et al., 2021b)

Background: Style transfer (1)

- **Motivation: Artistic image style transfer** (Gatys et al., 2015)
  - Given an image, change its style to the style of another image.

Vincent van Gogh

Background: Style transfer (2)

- **Motivation: Artistic image style transfer** (Gatys et al., 2015)
  - Given an image, change its style to the style of another image.

Vincent van Gogh

  - Idea. Learn what varies in one image (content) and what stays similar (style).

Background: NLG via text style transfer

- **Natural language style**
  - A specific choice of words of a particular group of people, genre, or similar

    Sometimes interpreted broadly, for example, sentiment polarities seen as styles

- **Two variations of text style transfer**
  1. Given a text, rewrite it to a text with similar content but different style.
  2. Given two texts, rewrite the content of one text in the style of the other.

  The first is usually done with neural models, trained on paired texts. The second resembles image style transfer.

Philosophical. *"The desire for exclusive markets is one of the most potent causes of war."*	Gothic. *"i am a desire of your exclusive markets, and that you are one of the most potent causes of your war in me."*

taken from Gero et al. (2019)

- **Specific problems of text style transfer**
  - Style is hard to isolate from content in text.
  - Violations of grammaticality and coherence are directly visible.
  - Text is not fully continuous, making abstraction of content and style harder.

Neural style transfer for bias modification (Chen et al., 2018)

- **Task**
  - Given a news headline with left (right) political bias on an event, modify the bias to right (left) while maintaining the event.
    Headlines of biased news articles are often claim-like statements.

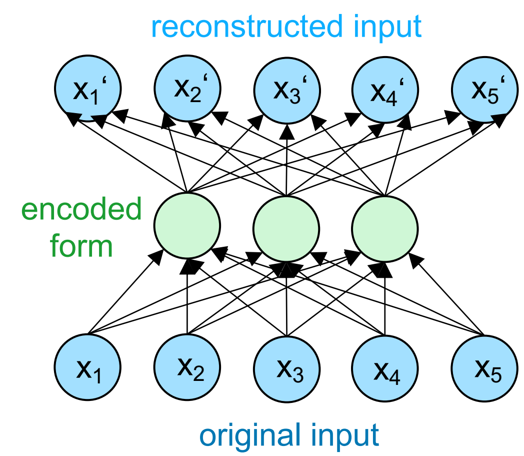
Trump is making a huge mistake on Jerusalem

Trump is right in recognizing Jerusalem as Israel's capital

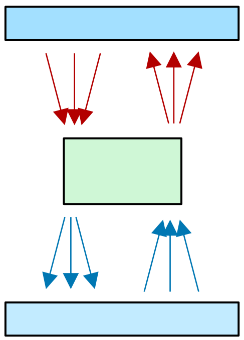
- **Research question**
  - Can bias modification be tackled as a style transfer task?
- **Data**
  - Headlines of 2196 pairs of left-/right-biased articles from *allsides.com.*
- **Approach in a nutshell**
  - Pre-train a neural sequence-to-sequence model on content of biased articles.
  - Fine-tune the model on generating one headline from the other.
  - Key idea. Use a *cross-aligned autoencoder*, to optimize the reconstruction of content in both directions.

Neural style transfer for bias modification: Approach

- **Background: Autoencoder**
  - An unsupervised neural network that learns to encode and decode input efficiently
    Network architectures of different complexity possible.
  - Encoding. Represent input in a compressed form.
  - Decoding. Reconstruct the original input from the compressed form.



- **Cross-aligned autoencoders for style transfer** (Shen et al., 2017)
  - Two autoencoders sharing the same encoded form, one for each style *A* and *B*
  - By simultaneously training on texts with similar content, the encoding represents content and decoding adds style.



- **Bias modification with cross-aligned autoencoders**
  - Represent input news headline with encoder of left bias.
  - Reconstruct encoded form of input with decoder of right bias. (or vice versa)

Neural style transfer for bias modification: Results

- **Manual evaluation**
  - Three annotators assessed 200 generated headlines in terms of event maintenance (Fleiss‘ $\kappa = 0.51$) and bias modification ($\kappa = 0.29$).

Results

- 63.5% have a correctly maintained event.
- 52.0% have a correctly modified bias.
- 41.5% are correct in both regards.

Obama accepts nomination, says his plan leads to a ”better place“

▼

Obama blasted re-election, saying it a ”very difficult“ to go down.

- **Observations**
  - Despite much room for improvement, the general idea seems to work.
  - With more data, the syntax generated by neural models gets much better.
  - The main challenge is the maintenance of semantics to the extent desired.

Lackluster Obama: change is hard, give me more time.

▼

Real GOP: debate is right, and more Trump

Counterargument synthesis: Discussion

- **Effective counterargument synthesis**
  - Most problems of general argument synthesis also come up here.
  - The input argument provides valuable information for countering it.
  - The challenge lies in opposing the stance while adhering to the given topic.

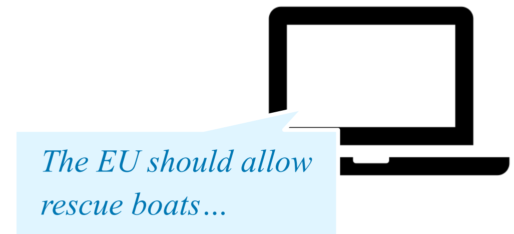
- **How to synthesize counterarguments in practice?**
  - Most approaches rely on some neural sequence-to-sequence model to connect the output to the input.
  - A common strategy is to retrieve and integrate units from existing arguments.
  - Counterargument generation is still rather experimental.

- **Why counterargument synthesis?**
  - Also here, increase of the capabilities of debating technologies
  - Raising awareness of potential counter-considerations for any argument
  - Sub-technologies may help to spot weak points in arguments

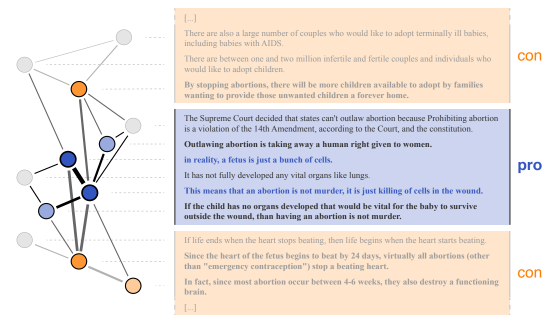
Next section: Conclusion

Conclusion

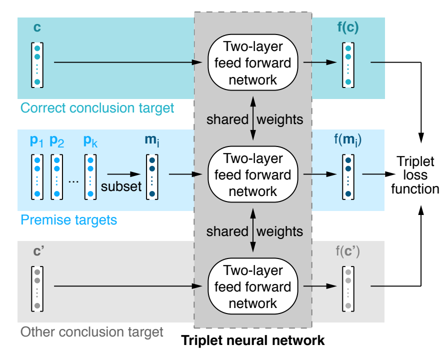
- **Argument generation**
  - Summarization of argumentative texts
  - Synthesis of arguments, their units, and longer texts
  - Synthesis of counterarguments



- **Summarization of argumentative texts**
  - Summaries may be based on one or multiple texts
  - Extractive and abstractive approaches exist
  - In a way, the gist of arguments needs to be found



- **Synthesis of arguments and counterarguments**
  - Composition of units to obtain new arguments
  - Neural approaches to generate new argumentative text
  - Style transfer to modify aspects of existing units



References

- **Akbik et al. (2018).** Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.
- **Al-Khatib et al. (2021).** Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou, and Benno Stein. Employing Argumentation Knowledge Graphs for Neural Argument Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, to appear, 2021.
- **Alshomary et al. (2020a).** Milad Alshomary, Shahbaz Syed, Martin Potthast, and Henning Wachsmuth. Target Inference in Argument Conclusion Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, to appear 2020.
- **Alshomary et al. (2020b).** Milad Alshomary, Nick Düsterhus, and Henning Wachsmuth. Extractive Snippet Generation for Arguments. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, to appear, 2020.
- **Alshomary et al. (2021a).** Milad Alshomary, Wei-Fan Chen, Timon Gurcke, Henning Wachsmuth. Belief-based Generation of Argumentative Claims. 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 224–233, 2021.
- **Alshomary et al. (2021b).** Milad Alshomary, Shahbaz Syed, Arkajit Dhar, Martin Potthast, Henning Wachsmuth. Counter-Argument Generation by Attacking Weak Premises. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1816–1827, 2021.
- **Bar-Haim et al. (2017a).** Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. Stance Classification of Context-Dependent Claims. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 251–261, 2017.

References

- **Bar-Haim et al. (2020).** Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. From Arguments to Key Points: Towards Automatic Argument Summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, to appear 2020.
- **Bilu and Slonim (2016).** Yonatan Bilu and Noam Slonim. Claim Synthesis via Predicate Recycling. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 525–530, 2016.
- **Bilu et al. (2019).** Yonatan Bilu, Ariel Gera, Daniel Hershcovich, Benjamin Sznajder, Dan Lahav, Guy Moshkowich, Anael Malet, Assaf Gavron, Noam Slonim. Argument Invention from First Principles. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1013–1026, 2019.
- **Carenini and Moore (2006).** Giuseppe Carenini and Johanna D. Moore. 2006. Generating and evaluating evaluative arguments. Artificial Intelligence, 170(11):925–952.
- **Chen et al. (2018).** Wei-Fan Chen, Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. Learning to Flip the Bias of News Headlines. In Proceedings of The 11th International Natural Language Generation Conference, pages 79–88, 2018.
- **Devlin et al. (2019).** Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pages 4171–4186, 2019.
- **Egan et al. (2016).** Charlie Egan, Advaith Siddharthan, and Adam Wyner. Summarising the points made in online political debates. In Proceedings of the Third Workshop on Argument Mining (ArgMining2016), pages 134–143, 2016.
- **El Baff et al. (2019).** Roxanne El Baff, Henning Wachsmuth, Khalid Al-Khatib, Manfred Stede, and Benno Stein. Computational Argumentation Synthesis as a Language Modeling Task. In Proceedings of the 12th International Conference on Natural Language Generation, pages 54–64, 2019.
- **Erkan and Radev (2004).** Günes Erkan and Dragomir R Radev. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research 22:457–479, 2004.

References

- **Gatys et al. (2015).** Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. CoRR abs/1508.06576, 2015.
- **Gero et al. (2019).** Katy Gero, Chris Kedzie, Jonathan Reeve, Lydia Chilton. Low Level Linguistic Controls for Style Transfer and Content Preservation. In Proceedings of the 12th International Conference on Natural Language Generation, pages 208–218, 2019.
- **Gurcke et al. (2021).** Timon Gurcke, Milad Alshomary, and Henning Wachsmuth. Assessing the Sufficiency of Arguments through Conclusion Generation. In Proceedings of the 8th Workshop on Argument Mining, pages 67–77, 2021.
- **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.
- **Hidey and McKeown (2019).** Christopher Hidey and Kathy McKeown. Fixed That for You: Generating Contrastive Claims with Semantic Edits. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1756–1767, 2019.
- **Hua and Wang (2018).** Xinyu Hua and Lu Wang. Neural Argument Generation Augmented with Externally Retrieved Evidence. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 219–230, 2018.
- **Hua et al. (2019).** Xinyu Hua, Zhe Hu, and Lu Wang. Argument Generation with Retrieval, Planning, and Realization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2661–2672, 2019.
- **Johnson and Blair (2006).** Ralph H. Johnson and J. Anthony Blair. 2006. Logical Self-defense. International Debate Education Association.

References

- **Lewis et al. (2019).** Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, 2020. 45–55, 2015.
- **Liu et al. (2019).** Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692, 2019.
- **Radford et al. (2018).** Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. OpenAI Blog, 2018.
- **Reisert et al. (2015).** Paul Reisert, Naoya Inoue, Naoaki Okazaki, Kentaro Inui. A Computational Approach for Generating Toulmin Model Argumentation. In Proceedings of the 2nd Workshop on Argumentation Mining, pages
- **Reiter and Dale (1997).** Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. Natural Language Engineering, 3(1):57–87.
- **Sato et al. (2015).** Misa Sato, Kohsuke Yanai, Toshinori Miyoshi, Toshihiko Yanase, Makoto Iwayama, Qinghua Sun, and Yoshiki Niwa. 2015. End-to-End Argument Generation System in Debating. In Proceedings of ACL-IJCNLP 2015 System Demonstrations, pages 109–114, 2015.
- **Schiller et al. (2020).** Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. Aspect-Controlled Neural Argument Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, to appear 2020.
- **Shen et al. (2017).** Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In Advances in Neural Information Processing Systems, pages 6833–6844, 2017.

References

- **Stab (2017).** Christian Stab. Argumentative Writing Support by means of Natural Language Processing, Chapter 5. PhD thesis, TU Darmstadt, 2017.
- **Stab and Gurevych (2017).** Christian Stab and Iryna Gurevych. Recognizing Insufficiently Supported Arguments in Argumentative Essays. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 980–990, 2017.
- **van der Lee (2019).** Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In Proceedings of the 12th International Conference on Natural Language Generation, pages 355–368, 2019.
- **Vaswani et al. (2017).** Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser. Attention Is All You Need. In 31st Conference on Neural Information Processing Systems, 2017.
- **Wachsmuth et al. (2018a).** Henning Wachsmuth, Shahbaz Syed, and Benno Stein. Retrieval of the Best Counterargument without Prior Topic Knowledge. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pages 241–251, 2018.
- **Wachsmuth et al. (2018b).** Henning Wachsmuth, Manfred Stede, Roxanne El Baff, Khalid Al-Khatib, Maria Skeppstedt, and Benno Stein. Argumentation Synthesis following Rhetorical Strategies. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3753–3765, 2018.
- **Wang and Ling (2016).** Lu Wang and Wang Ling. Neural Network-Based Abstract Generation for Opinions and Arguments. In: Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics, pages 47–57, 2016.
- **Zukerman et al. (2000).** Ingrid Zukerman, Richard McConachy, Sarah George. Using Argumentation Strategies in Automated Argument Generation. In INLG’2000 Proceedings of the First International Conference on Natural Language Generation, pages 55–62, 2000.