The international journal of science / 18 March 2021

index
Asia Pacific

nature

THE SILICON SOAPBOX

AI system goes head-to-head with humans in competitive debates

Zeroing in
Replace vague claims with rigorous plans to cut emissions

Almost blue
Modelling the path towards a sustainable blue economy

Defensive move
Exercise helps prompt production of immune cells in bone marrow

$10.00US $12.99CAN
11>
0 71486 03070 6

Article

An autonomous debating system

https://doi.org/10.1038/s41586-021-03215-w

Received: 19 May 2020

Accepted: 8 January 2021

Published online: 17 March 2021

Noam Slonim[1]✉, Yonatan Bilu[1], Carlos Alzate[2], Roy Bar-Haim[1], Ben Bogin[1], Francesca Bonin[2], Leshem Choshen[1], Edo Cohen-Karlik[1], Lena Dankin[1], Lilach Edelstein[1], Liat Ein-Dor[1], Roni Friedman-Melamed[1], Assaf Gavron[1], Ariel Gera[1], Martin Gleize[2], Shai Gretz[1], Dan Gutfreund[1], Alon Halfon[1], Daniel Hershcovich[1], Ron Hoory[1], Yufang Hou[2], Shay Hummel[1], Michal Jacovi[1], Charles Jochim[2], Yoav Kantor[1], Yoav Katz[1], David Konopnicki[1], Zvi Kons[1], Lili Kotlerman[1], Dalia Krieger[1], Dan Lahav[1], Tamar Lavee[1], Ran Levy[1], Naftali Liberman[1], Yosi Mass[1], Amir Menczel[1], Shachar Mirkin[1], Guy Moshkowich[1], Shila Ofek-Koifman[1], Matan Orbach[1], Ella Rabinovich[1], Ruty Rinott[1], Slava Shechtman[1], Dafna Sheinwald[1], Eyal Shnarch[1], Ilya Shnayderman[1], Aya Soffer[1], Artem Spector[1], Benjamin Sznajder[1], Assaf Toledo[1], Orith Toledo-Ronen[1], Elad Venezian[1] & Ranit Aharonov[1]

Artificial intelligence (AI) is defined as the ability of machines to perform tasks that are usually associated with intelligent beings. Argument and debate are fundamental capabilities of human intelligence, essential for a wide range of human activities, and common to all human societies. The development of computational argumentation technologies is therefore an important emerging discipline in AI research[1]. Here we present Project Debater, an autonomous debating system that can engage in a competitive debate with humans. We provide a complete description of the system's architecture, a thorough and systematic evaluation of its operation across a wide range of debate topics, and a detailed account of the system's performance in its public debut against three expert human debaters. We also highlight the fundamental differences between debating with humans as opposed to challenging humans in game competitions, the latter being the focus of classical 'grand challenges' pursued by the AI research community over the past few decades. We suggest that such challenges lie in the 'comfort zone' of AI, whereas debating with humans lies in a different territory, in which humans still prevail, and for which novel paradigms are required to make substantial progress.

Recent years have seen substantial progress in developing language models that adequately perform language understanding tasks[2–4]. Such tasks lie on a continuum of complexity. For simpler tasks, focused on specific linguistic phenomena such as predicting the sentiment of a given sentence[5], state-of-the-art systems often present excellent results[6]. On more complex tasks, such as automatic translation[7], automatic summarization[8] and dialogue systems[9], automatic systems still fall short of human performance. The task of holding a debate is positioned further along this complexity continuum. Debating represents a primary cognitive activity of the human mind, requiring the simultaneous application of a wide arsenal of language understanding and language generation capabilities, many of which have only been partially studied from a computational perspective (as separate tasks), and certainly not in a holistic manner[1,10]. Therefore, an autonomous debating system seems to lie beyond the reach of previous language research endeavours. Here, we describe such a system in full, and report results suggesting that this system can perform decently in a debate with a human expert debater.

The development of this system, referred to as Project Debater (https://www.research.ibm.com/artificial-intelligence/project-debater/), started in 2012, aiming to eventually demonstrate its capabilities in a live debate with a champion human debater. We defined a debate format which is a simplified version of the parliamentary debate style commonly used in academic competitive debates. Once the resolution—called the 'debate motion'—is announced, each side has 15 min of preparation time. Next, both sides alternate, delivering an opening speech of up to 4 min, a second speech of up to 4 min and closing statements of up to 2 min (Fig. 1). Speeches are typically composed of arguments supporting the speaker's position as well as arguments rebutting those raised by the other side. The audience votes on the motion before and after the debate, and the contestant who was able to pull more votes to their side is declared the winner. The official debut of Project Debater took place on 11 February 2019 (https://www.youtube.com/watch?v=m3u-1yttrVw), debating with H. Natarajan (a widely recognized debate champion, who was a grand finalist at the 2016 World Universities Debating Championships and winner of the European Universities Debating Championship in 2012) on the motion of whether preschool should be subsidized. The focus of this paper is to describe the system and its results across a wide range of topics, and not this specific event. Nonetheless, it is important to note that this motion was never included in the training data used for the development of the system. A full transcript of this debate, including information that elucidates the system's operation throughout, and the results of the audience vote, is provided in Supplementary Information section 11. Transcripts are also provided there for two additional debates held in June 2018 in front of a smaller audience.

[1]IBM Research AI, Haifa, Israel. [2]IBM Research AI, Dublin, Ireland. ✉e-mail: noams@il.ibm.com

	Pre-debate: both sides receive the motion and prepare	15 min
	Moderator introduces the motion to the audience	
Opening speeches	Project Debater delivers the 'government' opening speech	4 min
	Human debater delivers the 'opposition' opening speech and replies	4 min
Second speeches	Project Debater offers rebuttal and additional points	4 min
	Human debater offers rebuttal and additional points	4 min
Summary speeches	Project Debater provides final rebuttal and closing statements	2 min
	Human debater provides final rebuttal and closing statements	2 min

Fig. 1 | Debate flow. Details of the debate format.

System architecture

Given the variety of tasks required to engage in a debate, it seems implausible to envision a monolith solution in the form of an end-to end system, such as a single neural model. Instead, our approach was to break the problem into modular tangible tasks pursued in parallel. Interestingly, at the time, some of these tasks had received relatively little attention from the relevant scientific communities. For instance, the tasks of context-dependent claim detection[11] and context-dependent evidence detection[12] were proposed and formulated in the context of our project, and have since become an active area of research in the computational argumentation community. In the following we concisely describe all major components of the system and how they interact with one another (Fig. 2). In Supplementary Information section 5 we provide additional details.

Project Debater is composed of four main modules: argument mining, an argument knowledge base (AKB), argument rebuttal and debate construction. The first two modules are the source of content for the debate speeches. Argument mining pinpoints arguments and counter-arguments that are relevant for the motion, within a large text corpus. The AKB contains arguments, counter-arguments and other texts that are relevant to general classes of debates rather than to a single motion; once given a motion, it finds the most relevant of those to use in the debate. The argument rebuttal module matches potential opposition claims coming from the first two modules against the actual speech of the opponent and generates potential responses based on the matching results. Finally, the debate construction module selects which of the texts suggested by the other modules will make it to the debate, and arranges them into a coherent narrative.

Argument mining

Argument mining is done in two stages. In an offline stage, a large corpus of some 400 million newspaper articles (from the LexisNexis 2011-2018 corpus, https://www.lexisnexis.com/en-us/home.page) is processed, breaking the articles into sentences, and indexing these sentences by the words therein, the Wikipedia concepts they refer to[13], the entities they mention[14,15], and pre-defined lexicon words[16]. In the online stage, once given a motion the system relies on this index to perform corpus-wide sentence-level argument mining[17], retrieving claims and evidence related to the motion (here, a 'claim' is a concise statement that has a clear stance towards the motion, and 'evidence' is a single sentence that clearly supports or contests the motion, yet is not merely a belief or a claim, but rather provides an indication of whether a relevant belief or a claim is true; see Supplementary Information section 8 for more details).

First, sentences with a high propensity for containing such arguments are retrieved using tailored queries. Next, neural models are used to rank these sentences according to the probability that they represent relevant arguments[16–20]. Finally, a combination of neural and knowledge-based methods is used to classify the stance of each argument towards the motion[21,22]. At this stage, the system also uses a topic-expansion component[23] to better encompass the scope of relevant arguments. That is, if the topic-expansion component successfully identifies additional concepts that are relevant to the debate, it requests the argument mining module to search for arguments mentioning these concepts as well. For example, in a motion debating the two-party system in the USA, the topic-expansion component may suggest searching for arguments regarding a multi-party system and integrating them into the speeches, with the correct stance towards the motion. The argument mining module also searches for arguments that support the other side, aiming to prepare a set of claims the opponent may use and evidence that may serve as responses. This set is later used by the rebuttal module.

The AKB

The AKB aims to formally capture the commonalities between different debates. For example, when debating whether to ban certain substances or activities, the system can exploit more general arguments relating to the emergence of a black market. Texts in the AKB contain principled arguments, counter-arguments, and commonplace examples that may be relevant for a wide range of topics. These texts are authored manually—or extracted automatically and then manually edited—and are grouped together into thematic classes. Given a new motion, the system uses a feature-based classifier to determine which of the classes are relevant to this motion[24]. All authored texts associated with a matched class can then potentially be used in a speech, and the system selects those that it predicts to be most relevant based on their semantic relatedness to the motion[25]. These texts include not only arguments but also inspiring quotes, colourful analogies, an appropriate framing for the debate, and more.

The AKB also contributes to the rebuttal module. Principled arguments are mapped to counter-arguments that rebut them. Hence, if the system determines that such a principled claim was alluded to by the opponent, it can respond using the corresponding counter-argument. In addition, the AKB contains several key sentiment terms that are common to debates—for example, the word 'harmful'—that are mapped to pattern-based authored responses, which can be used to rebut arguments that focus on such a term.

Argument rebuttal

For argument rebuttal, the system first compiles a list of claims that might be mentioned by the opponent, termed 'leads', using (1) the argument mining module; (2) the AKB module; and (3) arguments extracted from iDebate (https://idebate.org/debatabase) in case the debate topic, or a close variant, is covered there. Next, IBM's Watson is used to convert the speech of the human opponent into text using its automatic speech-to-text service for custom language and custom acoustic models (https://www.ibm.com/cloud/watson-speech-to-text).

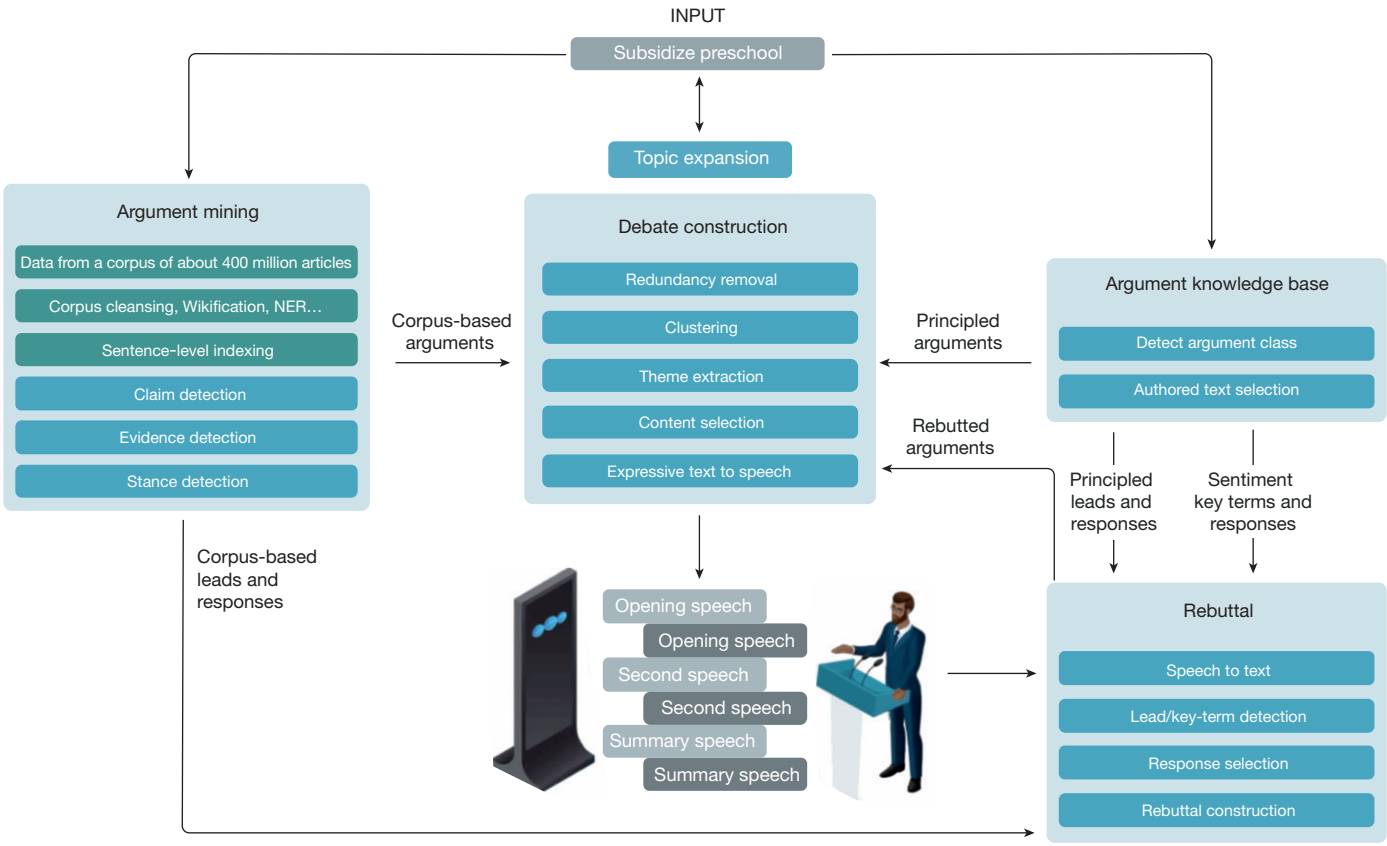

Fig. 2 | System architecture. Description of Project Debater components. Offline analysis is shaded in green; online analysis is shaded in blue. NER stands for named entity recognition.

A neural model splits the text obtained into sentences and adds punctuation[26]. Next, dedicated components aim to determine which of the pre-identified claim 'leads' were indeed stated by the opponent[27–29], and propose a rebuttal. AKB claims are rebutted with arguments listed in the AKB as rebutting them. Claims coming from argument mining are rebutted with mined evidence texts that mention similar concepts and are predicted to oppose the opponent's stance. For the few motions that match iDebate data, claims are rebutted with text based on the response listed in this resource. In addition to this claim-based rebuttal, key sentiment terms from the AKB are identified and serve as a cue for a simple form of rebuttal.

Debate construction

Finally, the Debate Construction module is a rule-based system that integrates cluster analysis. After removing arguments that are predicted to be redundant, the remaining arguments are clustered[30] according to their semantic similarity[25,31]. For each cluster, a theme is identified, which is a Wikipedia concept (such as 'poverty') that is statistically enriched in the cluster's arguments, and is used to introduce the respective paragraph. The system then selects the content that will be included in the debate: a salient subset of clusters for the speeches and the arguments per cluster, aiming to keep a diverse set of high-quality arguments[20] that reflect the cluster's theme. Next, various text-normalization and rephrasing techniques are applied to enhance fluency, and finally each speech is generated paragraph-by-paragraph using a pre-defined template. For example, the opening speech starts with the system greeting the opponent (this greeting was scripted for live events and presented to the audience as such), framing the motion, presenting general arguments derived from the AKB module, moving on to a brief introduction of the main points, followed by more specific arguments derived from the argument mining module and arranged into paragraphs via the clustering process. To vocalize the generated speech, the system uses an expressive text-to-speech service that was developed to suit argumentative content[32,33].

Evaluation and results

Evaluating the performance of a debate system is challenging because there is no single agreed-upon metric with which to determine a debate winner. In public debates, voting by the audience before and after the debate can determine the 'winning' side. But this approach has inherent limitations. First, if the pre-debate vote is highly unbalanced, the burden on the leading side is correspondingly higher (for example, in the pre-debate vote during the February 2019 debut, 79% of audience were in favour of subsidizing preschool, whereas only 13% were against it, so Project Debater had a potential of 21% of the audience to convince, whereas H. Natarajan had a potential of 87%). Further, voting naturally involves personal opinions, and is likely to be affected by various factors that are difficult to quantify and control. Moreover, producing a live debate with an impartial large audience is complicated, and producing many such debates is even more so. Nonetheless, a reliable estimation is essential in order to evaluate the overall performance of the system, to compare it to various baselines and to track its progress over time.

Comparison to baseline systems

We are unaware of any existing automatic method beyond Project Debater that can participate in a full debate. Hence, we compared Project Debater to other methods on the more limited task of generating an opening speech, which is clearly the first step any debating system should be capable of. We selected 78 motions to estimate the performance when a new, unknown motion is presented (Supplementary

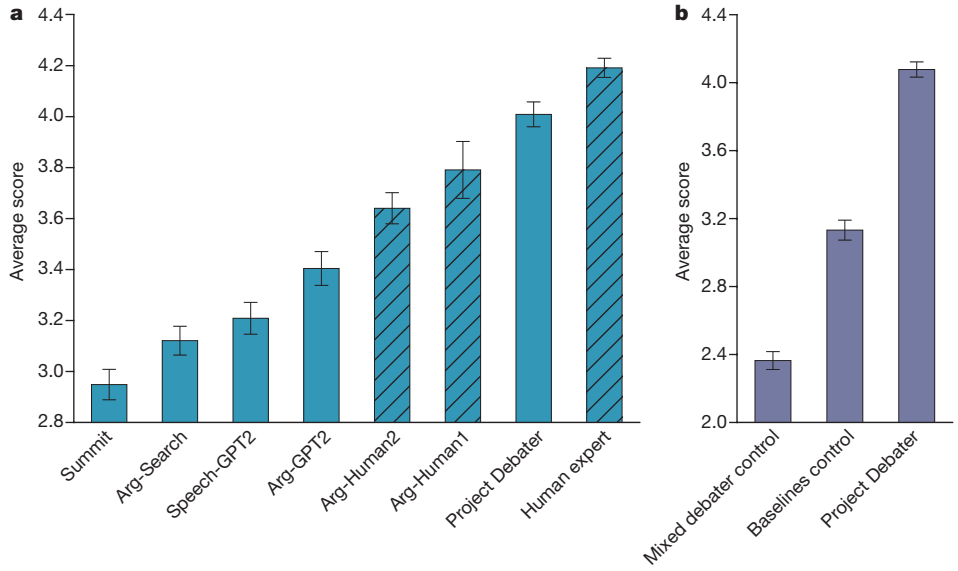

Fig. 3 | Evaluation of Project Debater. a, Comparison to baseline systems. Bars denote the average score, where 5 denotes 'Strongly Agree', and 1 'Strongly Disagree' with the statement 'This speech is a good opening speech for supporting the topic'. Striped bars indicate systems in which the speeches were generated by a human or relied on manually curated arguments. **b**, Evaluation of the final system. 'Project Debater' depicts the results when S1 and S3 are generated by Project Debater. In 'Mixed Debater Control', the third speech was an S3 generated by Project Debater but for a different motion. In 'Baselines Control', both S1 and S3 were opening speeches selected from one of the fully automatic baseline systems. Bars denote the average score, where 5 denotes 'Strongly Agree', and 1 'Strongly Disagree' with the statement 'The first speaker is exemplifying a decent performance in this debate'. In both panels, error bars denote the 95% confidence interval of the mean based on bootstrapping. The detailed labelling results are available in the Supplementary Information. Project Debater scores are significantly higher than the scores of all baselines and controls, and significantly lower than the scores of the Human Experts ($P < 0.05$ for both). For details on the statistical analysis see Supplementary Information section 7.

Information section 4), and considered nine opening speeches per motion, as follows. There were three speeches generated in full by (1) Project Debater, (2) a multi-document summarization system[34] (denoted 'Summit'), and (3) a fine-tuned GPT-2 language model[4] (denoted 'Speech-GPT2'). There were four speeches generated by concatenating arguments that were (1) generated via GPT-2 (denoted 'Arg-GPT2'), (2) extracted via ArgumenText[35] (denoted 'Arg-Search'), (3) authored by humans[36] (available for only 23 motions, and denoted 'Arg-Human1'), and (4) retrieved by Project Debater's argument mining module, which were further manually curated by humans to ensure that they represent valid arguments in the correct stance[17] (denoted 'Arg-Human2'). We note that the latter two speches rely on human annotation, providing challenging baselines. Finally, we included two opening speeches delivered by human expert debaters, reflecting the full human performance on this task (available for only 77 motions, and denoted 'Human Expert'). The nine speeches were presented in a random order, with no indication of the speech origin, to crowd annotators that had exhibited good performance on previous annotation tasks. We used a scenario-based approach, asking annotators to imagine themselves as part of a debate audience and to indicate to what extent they agree with several statements about the speech. Each speech was reviewed by 15 annotators. Figure 3a depicts the degree of agreement with the statement 'This speech is a good opening speech for supporting the topic', where 5 denotes strong agreement. The Project Debater results clearly outperform all baselines, and are rather close to the human expert scores.

Evaluation of the final system

For evaluation beyond the opening speech we used the same set of 78 motions, and again asked the selected group of crowd annotators to imagine themselves as part of a debate audience, but in this case they read three speeches, without knowing their origin. The three speeches were an opening speech in support of the motion, denoted S1; an opening speech contesting the motion, recorded by a human expert debater, denoted S2 (given that this speech is solely used as an anchor for the system's response, it was recorded as a response to an opening speech supporting the motion recorded by a different human debater); and finally another speech supporting the motion, denoted S3, which in principle should include a rebuttal to S2 as well as further arguments supporting the motion.

Each set of three speeches S1–S3 was reviewed by 20 annotators, asked to indicate to what extent they agree with the statement 'The first speaker is exemplifying a decent performance in this debate', focusing only on S1 and S3. Since we are unaware of any baseline method that can participate in a full debate, in Fig. 3b we depict the results of Project Debater versus two simple controls, designed mainly to verify the validity of the labelling. In all but three motions the average score of Project Debater was above the neutral 3 and for 50 out of 78 motions the average score was ≥4, suggesting that in at least 64% of the motions, the crowd annotators perceived Project Debater as demonstrating 'decent performance' in the debate.

It is worth noting the inherent challenges in evaluating the system, which are not completely overcome—the evaluation is only partial because annotators consider only S1 and S3, and the comparison is with simple controls rather than the performance of an experienced debater participating in a full debate. Indeed, even relying on annotators who read the text, rather than on an audience during a live debate, is a compromise.

The above two evaluations are of the system at the end of its development. However, working towards a 'grand challenge' event, it was important to track the progress of the system over time. To this end we performed a periodic evaluation analogous to the one described above, showing a clear improvement over the years. Finally, using the same set of motions for evaluation over a long time period raises the concern of gradually over-fitting to this set. To address this concern we performed an additional evaluation over an independent set of 36 motions, showing that the magnitude of over-fitting in our results—if present—is small (Supplementary Information section 7.4).

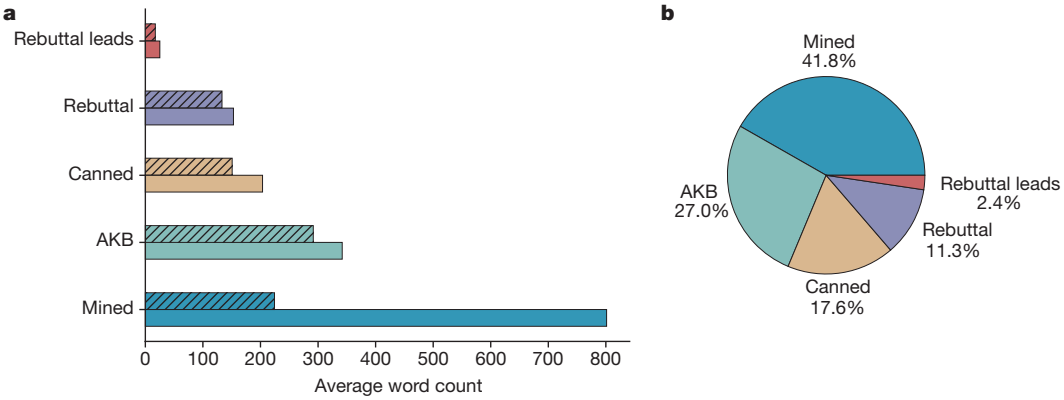

Fig. 4 | Content type analysis. a, Average word count of 5 content types, that cover the entire system output, in the 'low' (striped bars) and 'high' (plain bars) motions: mined arguments; arguments coming from the AKB; rebuttal; rebuttal leads; and conventional canned text. The average word counts are calculated for the 11 and 12 low and high motions, respectively, in the second evaluation set. **b**, Relative distribution of content types across all speeches in the 78 motions in the first evaluation set. See Supplementary Information section 10 for more details.

In-depth analysis

To gain more insights, we further analysed the results over these 36 motions. The errors are roughly divided into local errors, affecting a specific content unit in a speech, and more extensive errors that propagate through multiple elements and affect the speech as a whole. The most common types of local errors were mistakes in classifying argument stance; elements that appear to be off-topic and do not fit into the overall speech narrative; and elements that are incoherent without additional context. In an extensive error, the same type of mistake recurs throughout a speech. For example, in one motion the identified AKB classes were not a good match, resulting in a large amount of AKB content in the speeches that was entirely off-topic. Other cases of extensive errors were of a more complex nature, illustrating the need for a nuanced and holistic understanding of context. For instance, for the motion 'We should increase the use of artificial insemination', the system output included arguments pertaining to artificial insemination in livestock, which may sound awkward in this debate; in another motion, 'We should not subsidize athletes', the system suggested multiple arguments about negative health outcomes that afflict athletes, while only partially addressing the core issue of whether subsidizing athletes is a desired policy approach.

We further divided these motions into three groups based on the evaluation scores given by in-house annotators—'high' (12 motions with a score above 3.5); 'medium' (11 motions between 3 and 3.5); and 'low' (11 motions below 3). Notably, extensive errors as described above occurred only in the 'low' group. In contrast, local errors appear to some extent in almost all of the evaluated motions, including those in the 'high' group. Otherwise, the most prominent difference between the groups was the amount of content in the three speeches. In terms of total word count, 'high', 'medium' and 'low' motions had an average of 1,496, 1,155 and 793 words, respectively. This hallmark of 'low' motions reflects the challenge of constructing a system that relies on the output of many components and is meant to generate a precision-oriented output over a wide variety of topics. Specifically, for the system to find relevant content, the motion's topic must be discussed in the corpus; and for a specific content unit to be included in the final output, it must pass multiple confidence thresholds, which are set to be strict, to ensure high precision. This, in turn, may result in much of the relevant content being filtered out. Correspondingly, generating several minutes of spoken language content that is relevant and to the point is a formidable task. Focusing on those motions that do have a reasonable amount of content, another salient property is the quality of the narrative framing, provided by AKB elements at the opening and closing of speeches. 'High' motions typically have framing elements that accurately capture the essence of a debate (for example, framing about the importance of privacy in 'We should end the use of mass surveillance'), whereas 'medium' ones tend to have a framing that is acceptable but less on point.

Finally, we analyse the word frequency of five content types, which cover the entire system output: mined arguments; arguments coming from the AKB; rebuttal; rebuttal leads; and conventional canned text. Figure 4a depicts that across all types we have less content for 'low' motions compared to 'high' ones, in line with our analysis above. The largest gap is in the mined content, further suggesting that high-quality output is associated with an abundance of relevant arguments in the examined corpus, pinpointed by the argument mining module. In addition, we examined the relative distribution of content types across all speeches in all 78 motions in our original evaluation set (Fig. 4b). Evidently, less than 18% of the content is conventional canned text, while the remaining content is contributed by the more advanced underlying system components.

Discussion

Research in AI and in natural language processing is often focused on so called 'narrow AI', consisting of narrowly defined tasks. The preference for such tasks has several reasons. They require less resources to pursue; typically have clear evaluation metrics; and are amenable to end-to-end solutions such as those stemming from the rapid progress in the study of deep learning techniques[37]. Conversely, 'composite AI' tasks—namely, tasks associated with broader human cognitive activities, which require the simultaneous application of multiple skills—are less frequently tackled by the AI community. Here, we break down such a composite task into a collection of tangible narrow tasks and develop corresponding solutions for each. Our results demonstrate that a system that properly orchestrates such an arsenal of components can meaningfully engage in a complex human activity, one which we presume is not readily amenable to a single end-to-end solution.

Since the 1950s AI has advanced in leaps and bounds, thanks, in part, to the 'grand challenges', in which AI technologies performed tasks of growing complexity. Often, this was in the context of competing against humans in games which were thought to require intuitive or analytic skills that are particular to humans. Examples range from chequers[38], backgammon[39], and chess[40], to Watson winning in Jeopardy![41] and Alpha Zero winning at Go and shogi[42].

We argue that all these games lie within the 'comfort zone' of AI, whereas many real-world problems are inherently more ambiguous and fundamentally different, in several ways. First, in games there is a clear definition of a winner, facilitating the use of reinforcement learning

techniques[39,42]. Second, individual game moves are clearly defined, and the value of such moves can often be quantified objectively (for example, see ref. [43]), enabling the use of game-solving techniques. Third, while playing a game an AI system may come up with any tactic to ensure winning, even if the associated moves could not be easily interpreted by humans. Finally, for many AI grand challenges, such as Watson[41] and Alpha Star[44], massive amounts of relevant structured data (for example, in the form of complete games played by humans) was available and imperative for the development of the system.

These four characteristics do not hold in competitive debate, which requires an advanced form of using human language, one with much room for subjectivity and interpretation. Correspondingly, often there is no clear winner. Moreover, even if we had a computationally efficient 'oracle' to determine the winner of a debate, the sheer complexity of a debate—such as the amount of information required to encode the 'board state' or to enumerate all possible 'moves'—prohibits the use of contemporary game-solving techniques. In addition, it seems implausible to win a debate using a strategy that humans can fail to follow, especially if it is the human audience which determines the winner. And finally, structured debate data are not available at the scale required for training an AI system. Thus, the challenge taken by Project Debater seems to reside outside the AI comfort zone, in a territory where humans still prevail, and where many questions are yet to be answered.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-021-03215-w.

1. Lawrence, J. & Reed, C. Argument mining: a survey. *Comput. Linguist*. **45**, 765–818 (2019).
2. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint at https://arxiv.org/abs/1810.04805 (2018).
3. Peters, M. et al. Deep contextualized word representations. In *Proc. 2018 Conf. North Am. Ch. Assoc. for Computational Linguistics: Human Language Technologies* Vol. 1, 2227–2237 (Association for Computational Linguistics, 2018); https://www.aclweb.org/anthology/N18–1202
4. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* **1**, http://www.persagen.com/files/misc/radford2019language.pdf (2019).
5. Socher, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)* 1631–1642 (Association for Computational Linguistics, 2013).
6. Yang, Z. et al. XLNet: generalized autoregressive pretraining for language understanding. In *Adv. in Neural Information Processing Systems (NIPS)* 5753–5763 (Curran Associates,2019).
7. Cho, K., van Merriënboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: encoder–decoder approaches. In *Proc. 8th Worksh. on Syntax, Semantics and Structure in Statistical Translation* 103–111 (Association for Computational Linguistics, 2014).
8. Gambhir, M. & Gupta, V. Recent automatic text summarization techniques: a survey. *Artif. Intell. Rev*. **47**, 1–66 (2017).
9. Young, S., Gašić, M., Thomson, B. & Williams, J. POMDP-based statistical spoken dialog systems: A review. *Proc. IEEE* **101**, 1160–1179 (2013).
10. Gurevych, I., Hovy, E. H., Slonim, N. & Stein, B. *Debating Technologies (Dagstuhl Seminar 15512)* Dagstuhl Report 5 (2016).
11. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E. & Slonim, N. Context dependent claim detection. In *Proc. COLING 2014, the 25th Int. Conf. on Computational Linguistics: Technical Papers* 1489–1500 (Dublin City University and Association for Computational Linguistics, 2014); https://www.aclweb.org/anthology/C14–1141
12. Rinott, R. et al. Show me your evidence—an automatic method for context dependent evidence detection. In *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing* 440–450 (Association for Computational Linguistics, 2015); https://www.aclweb.org/anthology/D15–1050
13. Shnayderman, I. et al. Fast end-to-end wikification. Preprint at https://arxiv.org/abs/1908.06785 (2019).
14. Borthwick, A. *A Maximum Entropy Approach To Named Entity Recognition*. PhD thesis, New York Univ. https://cs.nyu.edu/media/publications/borthwick_andrew.pdf (1999).
15. Finkel, J. R., Grenager, T. & Manning, C. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. 43rd Ann. Meet. Assoc. for Computational Linguistics* 363–370 (Association for Computational Linguistics, 2005).
16. Levy, R., Bogin, B., Gretz, S., Aharonov, R. & Slonim, N. Towards an argumentative content search engine using weak supervision. In *Proc. 27th Int. Conf. on Computational Linguistics (COLING 2018)* 2066–2081, https://www.aclweb.org/anthology/C18-1176.pdf (International Committee on Computational Linguistics, 2018).
17. Ein-Dor, L. et al. Corpus wide argument mining—a working solution. In *Proc. Thirty-Fourth AAAI Conf. on Artificial Intelligence* 7683–7691 (AAAI Press, 2020).
18. Levy, R. et al. Unsupervised corpus-wide claim detection. In *Proc. 4th Worksh. on Argument Mining* 79–84 (Association for Computational Linguistics, 2017); https://www.aclweb.org/anthology/W17–5110
19. Shnarch, E. et al. Will it blend? Blending weak and strong labeled data in a neural network for argumentation mining. In *Proc. 56th Ann. Meet. Assoc. for Computational Linguistics* Vol. 2, 599–605 (Association for Computational Linguistics, 2018); https://www.aclweb.org/anthology/P18–2095
20. Gleize, M. et al. Are you convinced? Choosing the more convincing evidence with a Siamese network. In *Proc. 57th Conf. Assoc. for Computational Linguistic*, 967–976 (Association for Computational Linguistics, 2019).
21. Bar-Haim, R., Bhattacharya, I., Dinuzzo, F., Saha, A. & Slonim, N. Stance classification of context-dependent claims. In *Proc. 15th Conf. Eur. Ch. Assoc. for Computational Linguistics* Vol. 1, 251–261 (Association for Computational Linguistics, 2017).
22. Bar-Haim, R., Edelstein, L., Jochim, C. & Slonim, N. Improving claim stance classification with lexical knowledge expansion and context utilization. In *Proc. 4th Worksh. on Argument Mining* 32–38 (Association for Computational Linguistics, 2017).
23. Bar-Haim, R. et al. From surrogacy to adoption; from bitcoin to cryptocurrency: debate topic expansion. In *Proc. 57th Conf. Assoc. for Computational Linguistics* 977–990 (Association for Computational Linguistics, 2019).
24. Bilu, Y. et al. Argument invention from first principles. In *Proc. 57th Ann. Meet. Assoc. for Computational Linguistics* 1013–1026 (Association for Computational Linguistics, 2019).
25. Ein-Dor, L. et al. Semantic relatedness of Wikipedia concepts—benchmark data and a working solution. In *Proc. Eleventh Int. Conf. on Language Resources and Evaluation (LREC 2018)* 2571–2575 (Springer, 2018).
26. Pahuja, V. et al. Joint learning of correlated sequence labelling tasks using bidirectional recurrent neural networks. In *Proc. Interspeech* 548–552 (International Speech Communication Association, 2017).
27. Mirkin, S. et al. Listening comprehension over argumentative content. In *Proc. 2018 Conf. on Empirical Methods in Natural Language Processing* 719–724 (Association for Computational Linguistics, 2018).
28. Lavee, T. et al. Listening for claims: listening comprehension using corpus-wide claim mining. In *ArgMining Worksh*. 58–66 (Association for Computational Linguistics, 2019).
29. Orbach, M. et al. A dataset of general-purpose rebuttal. In *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing* 5595–5605 (Association for Computational Linguistics, 2019).
30. Slonim, N., Atwal, G. S., Tkačik, G. & Bialek, W. Information-based clustering. *Proc. Natl Acad. Sci. USA* **102**, 18297–18302 (2005).
31. Ein Dor, L. et al. Learning thematic similarity metric from article sections using triplet networks. In *Proc. 56th Ann. Meet. Assoc. for Computational Linguistics* Vol. 2, 49–54 (Association for Computational Linguistics, 2018); https://www.aclweb.org/anthology/P18–2009
32. Shechtman, S. & Mordechay, M. Emphatic speech prosody prediction with deep Lstm networks. In *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* 5119–5123 (IEEE, 2018).
33. Mass, Y. et al. Word emphasis prediction for expressive text to speech. In *Interspeech* 2868–2872 (International Speech Communication Association, 2018).
34. Feigenblat, G., Roitman, H., Boni, O. & Konopnicki, D. Unsupervised query-focused multi-document summarization using the cross entropy method. In *Proc. 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval* 961–964 (Association for Computing Machinery, 2017).
35. Daxenberger, J., Schiller, B., Stahlhut, C., Kaiser, E. & Gurevych, I. Argumentext: argument classification and clustering in a generalized search scenario. *Datenbank-Spektrum* **20**, 115–121 (2020).
36. Gretz, S. et al. A large-scale dataset for argument quality ranking: construction and analysis. In *Thirty-Fourth AAAI Conf. on Artificial Intelligence* 7805–7813 (AAAI Press, 2020); https://aaai.org/ojs/index.php/AAAI/article/view/6285
37. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
38. Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM J. Res. Develop*. **3**, 210–229 (1959).
39. Tesauro, G. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Comput*. **6**, 215–219 (1994).
40. Campbell, M., Hoane, A. J., Jr & Hsu, F.-h. Deep Blue. *Artif. Intell*. **134**, 57–83 (2002).
41. Ferrucci, D. A. Introduction to "This is Watson". *IBM J. Res. Dev*. **56**, 235–249 (2012).
42. Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).
43. Coulom, R. Efficient selectivity and backup operators in Monte-Carlo tree search. In *5th Int. Conf. on Computers and Games* inria-0011699 (Springer, 2006).
44. Vinyals, O. et al. Grandmaster level in Starcraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).

To order reprints, please contact:
Tel +1 212 726 9278; reprints@us.nature.com
Printed by The Sheridan Press