Association for Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
August 2018

Artificial Intelligence in Politics

An Interview with Sven Körner and Mathias Landhäußer of thingsTHINKING

by Walter Tichy

Editor's Introduction

Natural language processing, an area of artificial intelligence (AI), has attained remarkable successes. Digital assistants such as Siri and Alexa respond to spoken commands, and understand several languages. Google has demonstrated a machine can call up a restaurant and make a reservation in a manner that is indistinguishable from a human. Automated translation services are used around the world in over a hundred languages. This interview discusses a new and surprising application of language processing in politics. Though the AI software analyzes texts in German, it could be adapted to any language. The underlying technology has wider applications in text analysis, including legal tech, contracting, and others. Here is a summary.

Artificial Intelligence in Politics

An Interview with Sven Körner and Mathias Landhäußer of thingsTHINKING

by Walter Tichy

The results of the German elections of 2017 forced political parties to form coalitions.[1] On February 11, 2018, the two largest parties (the alliance of the Christian Democratic Union and Christian Social Union, commonly known as CDU/CSU, and the Social Democratic Party, or SPD) finally announced the completion of their written coalition agreement. Instantly, a debate about whether the agreement was commensurate with the number of seats the parties won began. (CDU/CSU had 33 percent of the votes, while SPD had 20 percent.) Had one party succeeded in placing more of its policies into the agreement, to the disadvantage of the other?

thingsTHINKING, a startup with ties to the Karlsruhe Institute of Technology (KIT), decided to let the computer answer this question. Its artificial intelligence (AI) software compared each party's program against the coalition agreement to check how many political goals made it into the agreement. This involved a sentence-by sentence comparison of each party's program with the coalition agreement. Doing this work manually would be a daunting task, since the CDU/CSU's program is 76 pages, the SPD's 116 pages, and the agreement itself is 176 pages long. The machine performed the comparison within minutes, with high accuracy.

In this interview, two of the founders of thingsTHINKING discuss the results produced by the machine, how their software works, and where else it could be applied. They also clarify whether their technology could help identify fake news or plagiarism.

(This interview has been slightly edited for clarity.)[2]

[1] Like other parliamentary political systems (common throughout much of the world), in Germany when no single party wins an outright majority of parliamentary seats, the different political parties must come together to form a coalition. In other words, two or more parties must agree to work together to create a majority. Then together, they form a government, designating the Federal Chancellor and ministers of key governmental departments (finance, foreign affairs, justice, etc.).

[2] We apologize for publishing this interview long after the election; peer reviews took time, especially since the interviewees had a start-up to run.

Walter Tichy (WT): What were the results of the computer analysis?

Sven Körner (SK): We let the computer analyze how much of the parties' political programs made it into the coalition agreement. The software identified 2-3 times more thematic relationships between the SPD's political program and the coalition agreement than with the CDU/CSU program, even though the SPD had fewer votes. Just like human beings, the machine does not look at the actual wording. Rather, it builds a meaning model of the sentences and is therefore able to compare semantics. For instance, "A runs away from B" and "B is trying to catch A" are similar to the machine (in the same context). Here is a graph of the results (see Figure 1).

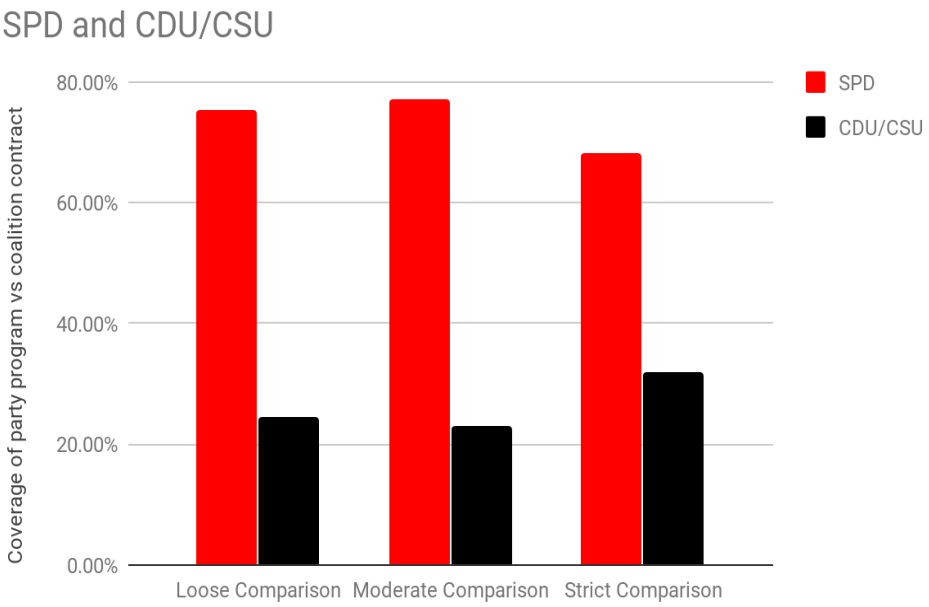

Figure 1. The graph shows the results of thingsTHINKING's analysis of the 2018 coalition agreement between Germany's two main political parties.

WT: The figure shows three comparisons. What do you mean by "loose," "moderate," and "strict" comparison—and which one is the actual result?

Mathias Landhäußer (ML): We can configure how strictly the software compares the documents. For example, when we tell the software to be extremely lax, it tells us that text fragments are "similar" if they are written in the same language no matter the content. When we tell it to be extremely strict, it only considers fragments to be "similar" when they are—in fact—identical. The bar plots in the figure correspond to three reasonable configurations that become more and more strict from left to right

WT: Can you give us some examples?

SK: Sure.

- CDU: "We want that all our children receive the best possible upbringing, education, and care, independent of their parents' origin and life situation."
- SPD: "We will eliminate the disadvantages of children of poor parents and provide equal participation."

The machine considers these statements[3] as similar in the given context.

WT: **How do you know the results are trustworthy?**

SK: As mentioned, we can configure the software to be more or less strict. We ran the analysis in different configurations (more than the three in the figure) and the ratios stayed roughly the same. Based on our experience, this is an indicator that the numbers are reliable. To cross-check the results we compared the political documents to unrelated texts from user manuals, Ph.D. theses, and other textual corpora and found no significant overlap.

ML: And we analyzed the texts from different viewpoints. When one considers the opposite question, "how much of the coalition agreement is in a party's program," we find more of it in the SPD program than in the CDU/CSU's. Also, we can cluster the party programs according to certain areas, subjects, and focal points and compare them to the coalition agreement. Often the media report that the two parties have grown more and more similar over the years. When we compared their programs to each other, we found exactly that: They are roughly one-third equal. Finally, we printed out all correspondences between party programs and the coalition agreement and deposited them with a trustee. Anyone interested can check the results.

[3] The sample statements were translated from German.

WT: How large is this set, and is anyone doing a manual check?

SK: The data is a total of 246 MB of reports, i.e., plain text. The top hits still sum up to more than 14MB of plain text and comparison values. Working through the set—and therefore the contracts—would take days.

Concerning the question if people are actually doing manual checks: yes, funny enough. We had requests from newspaper readers who wanted the full reports. But we never heard back from them. Also, there have been universities that—at least from Twitter statements—were working on replicating our results. After we ran the first tests, we took 150 random samples and gave them to a handful of lawyers to double-check the samples. It turns out we were not quite able to determine the machine's error rate due to differences in the human assessments (lawyers have differing opinions). In short, there are a number of paragraphs and statements in the coalition contract that might or might not be in the corresponding party's program. It all comes down to how freely you interpret the meaning. And as we say, the machine cannot perform better than humans, only faster.

WT: How important or substantial are the points that the parties were able to negotiate? Couldn't it be the case that one party concedes a few negligible points in order to win a large concession elsewhere?

ML: These are questions we cannot answer. Our study cannot rate the quality of the negotiations rather than (just) the quantitative aspects. We found there is more of one party in the contract even though that party won roughly 50 percent fewer votes during the election. Whether the negotiated clauses are of high or little value (and to whom) is something we cannot decide. Personally, I doubt that the SPD's negotiators outwitted Chancellor Merkel. I rather believe the parties had different priorities and that quantity does not translate to quality or importance. The software cannot help there, either: We based our analysis on common world knowledge, and the machine has no societal value system to decide what points are more important than others. To make a qualified statement regarding your question, we'd have to include specific knowledge about politics, ethics, and even rhetorics.

SK: This is what we do in our customer projects where the machine receives additional input (or bias) from the customer. This will help it rate clauses, texts, and paragraphs. In this specific case, the study stops at quantitative results since we would have to configure two systems, one for each party. That is due to the fact that each party has its own emphasis. These are the aspects that are impossible to grasp mathematically since they are not based on actual ground

truth, but on perceived truth of each of the parties. What is possible—but certainly not during a lunch break, as in this case—is to configure the machine so it could be a sidekick for contract negotiations. The machine would have the customer's bias and could be adapted for each customer.

WT: **We've spent quite some time on the results now, can you please explain how the system works?**

SK: Our system compares each sentences of an input document (in this case a party program) to each sentence of a reference document (in this case the coalition agreement). The analysis then connects text passages from the input document to text passages from the reference document if they are thematically related.

This is usually only the first step, and sufficient for analyzing the coalition agreement. In a second step we can also extract specific data from the text, and in a third step the software could reason about this data. Given a proper configuration, the system could answer your questions regarding the quality of negotiation results.

WT: **How does the software make the connection? This sounds a bit like searching for keywords.**

SK: Keyword spotting would be one way to do this—but no. Our research at KIT and many real-world use cases showed that working with natural language cannot be handled by a single technical approach. For instance, there have been discussions over the years whether symbolic (ontologies, inference, search) or sub-symbolic (neural networks, SVMs, LDA, etc.) approaches are best for machines to handle natural language. Well, we think it's a wild combination of those, plus a little "spice" to make it all tasty once you cook it. We would agree with Gary Marcus' "whatever works" approach. In short, we are not religious regarding technology. Our software uses a combination of different methods to identify semantic similarities in texts depending on the language, style, length, etc. When we enter a special (constrained) domain, say non-disclosure agreements, we can train additional, special-purpose models that take the peculiarities of the domain into account. If enough data exists, training can be done in an unsupervised way in what we call a "bottom-up" fashion [1].

ML: I'll explain a basic method that uses word embeddings. Word embeddings map a word to a (high dimensional) vector of real numbers. The vector space and the vectors are constructed in

such a way that vectors of semantically related words are geographically close. For example, "king" and "queen" are close to each other and "airplane"is in a different location together with other means of transportation. Word2vec is a popular toolkit for learning word embeddings released by Google in 2013, but the underlying idea is much older. Firth was one of the first linguists to express the idea that words have a semantic relationship when they occur in similar contexts often [2]. We can think of the word embeddings as semantic fingerprints that abstract from the actual wording: the fingerprints of "automobile" and "car" are almost identical. One can combine the vectors of a sentence's words to get a vector for the sentence. Once we have done this, identifying semantically similar spots boils down to comparing the fingerprints. Then we only have to aggregate the information from sentences to text passages. But this goes way too deep into the details.

WT: **What about negations? Suppose one document says: "We will raise the minimum wage"; the other says: "We will *not* raise the minimum wage." Won't these two sentences be treated as equivalent with word embeddings?**

ML: Classic word embeddings will indicate a very high similarity for the two statements—after all, the statements are literally almost identical. But when it comes to semantics, the difference couldn't be bigger. Negative words (such as "not" in your example) could have specific features but their vectors are very similar to their positive counterparts (for example, "absolutely"). So from Word2vec's perspective, the words are just more or less similar. Only if we had used an extremely strict setting during the analysis would the difference in the sentence vectors have been detected with word embeddings. In general, that is an issue. In the specific analysis of the political documents it was not: In the party programs and the coalition contract, positive speech is preferred—after all this is politics. Therefore the problem did not arise—at least we did not notice such examples during our inspection of the results. But getting to the core of your question, if one wants to interpret negations or potential conflicts as in your example (I'd rather not call it contradictions, because these are too hard to find in general), one would cater to the specifics of that problem. One would enrich the training data with additional information from part-of-speech taggers, or parsers, and then train a second neural network to evaluate whether a sentence has a positive or negative meaning (not sentiment, that's a different story). This way the first analysis would identify the statements are almost identical, and the second would tell you the first one is a positive statement and the second a negative one. You could use this approach to identify where your party's policy was not only discarded (because it is missing in the coalition agreement) but even contradicted (because it is in the agreement but with a negated meaning).

WT: **Isn't it easy to fool the special treatment of negations? Suppose the second sentence said: "We will keep the minimum wage as it is." Then you need some semantic processing to find out that raising something is not the same as leaving it unchanged.**

ML: Depending on the training data—in our case world knowledge —this case might actually be easier to solve than the negation problem. One part of our system learns meaning from reading many texts. The phrases "to keep" and "raising something" would have different meanings. This challenge is similar to the above example, where "the best possible upbringing" and "eliminate the disadvantages" is interpreted as semantically similar—independent of the actual words used.

WT: Are plagiarism detectors such as TurnItIn or JPlag similar to what you are doing?

ML: No. A plagiarism detector like TurnItIn looks for exact copies of phrases in texts. With word embeddings, the whole point is that the words need not be the same. So our software would produce far too many false positives.

JPlag compares programs. It produces an internal representation called an abstract-syntax tree for each program and searches for similar sub-trees. It eliminates identifiers (the "words") entirely. Instead, it detects structural similarity, even if the plagiarizer renamed all identifiers or translated them to another language. So the techniques are quite different.

WT: **Could you detect fake news?**

SK: That's a tough one! If we had alternative texts for the same topic, we could. For example, the UN recently published a report that 18.5 million Americans live in extreme poverty. In a rebuke, U.S. officials said there appear to be only 250,000 Americans in extreme poverty. A neural net trained for finding discrepancies could detect this, but the software couldn't say which claim is correct. It does not know how extreme poverty is defined (nor do most readers).

A more complex example: President Trump tweeted "Crime in Germany is up 10% plus" (June 9, 2018). This claim has no basis in fact. To debunk it, one would have to go to the Federal Statistics Office of Germany. There you would find that crime in Germany is currently at a 26-year low. From this example we see fact checking is not simply a matter of comparing texts. Note that there is also a translation problem lurking here.

A better approach might be to train a classifier to identify fake news using the following indicators: the source of the news, the geolocation of the reporter, the medium being used, the political leanings of source, reporter and medium, the choice of words, the topic, and perhaps the history of the news item. A neural net should be able to call out fake news with adequate accuracy, but the final check would still have to be done by the human. In social nets you could also consider the reputation of the network itself and the flow of information. Social network providers are working on this problem.

WT: **Where else could one use text comparison, or more general, text analysis, and what techniques are needed?**

SK: Here are some examples:

- Extract information from rental contracts and use that in tax returns (imagine real estate companies with thousands of contracts).
- Analyze tens of thousands of requirements documents of a company to see (1) whether requirements contain weak or unclear expressions, (2) whether there are duplicates in the database of requirements, (3) whether a new requirement in a customer's loose verbiage matches an existing requirement (some companies have precise standards for formulating requirements), and (4) whether something is missing.
- Scan numerous legal cases for relevance to a current lawsuit or identify problematic clauses in legal documents, for example non-disclosure agreements. In essence, support paralegals.
- Process documents in large data rooms in due diligence and mergers and acquisition cases. In such cases, there is more information in the documents than explicitly stated, i.e., the interpretation of the documents is really important. This is where semantics are a solution.
- Search confiscated data. Here, the searched-for items are weakly specified, and keyword search performs poorly.

The text analysis is only the first of a three-tier approach:

1. Our semantic similarity layer uses semantic generalization to compare contents on different levels of abstraction. It compares semantic models to find differences and

similarities among corresponding models. Once you found the right spots in documents and unstructured information, it is time to move to the next step.

2. With semantic extraction, we extract relevant and useful information from unstructured data and represent it in a structured format for further analysis. This is important because once you understand data, there are always other systems that need that data in a structured way. For instance, in the auditing space we process rental contracts and compare the extracted values to the information that's stored in the accounting systems to verify or double-check the recorded entries. Extracting information semantically has two benefits: First, it more or less ignores form. Therefore different documents of the same type can be processed and data extracted independently of their structure, wording, or layout. The second benefit is more often than not, different expressions with the same meaning are used. Also, once taught its domain, the machine detects if expected information seems to be missing. The reason could be that (a) it did not discover the information or (b) it is actually missing. Either way, it can provide what we call a "finding," which can then be used for interacting with the user, for instance by posing a follow-up question tailored to the specific use case. Of course, this interaction could be used for improving the machine further though you'd have to be careful—knowledge is not distributed equally—which would then again lead to wrong interpretations or at least bias.

3. The third layer leverages semantic knowledge to make decisions that today only humans can take. This is where GOFAI (good old-fashioned AI) approaches including knowledge graphs come into play. They were pretty uncool in 2016 and 2017, but are now having a renaissance due to the number of mishaps and limitations of data-driven approaches in real-world projects. Even Paul Allen's AI2 went back to it recently [4].

4. In this step, the machine can—depending on the domain of application—reason on the information extracted in step 2. This step is important for all challenges that do not have enough training data—be it due to lack of data or legal restrictions. The legal field is a good example for that. For large-scale repetitive problems supervised-learning approaches work, e.g., for NDAs or flight right litigation. It simply does not work for more complex scenarios. This is where the "instruction" of our solution comes into play. Essentially, you explain to the machine the specific domain it operates in, it will do the reasoning from there and draw its own conclusions. This might take as long as it takes to teach a human to do the same job—it just scales vastly better. Therefore we encourage to not instruct everything that's possible, but to tackle the main efforts and mitigate

those with machine help. The special cases will be left for humans, and probably for a while.

WT: **Will the machine catch everything it is supposed to catch?**

ML: No, but the great thing about this approach is the machine knows when it doesn't know. That means, it won't work through a 100-page document and tell you it finished with 87 percent accuracy, essentially not telling you which 13 pages it messed up. No, it would tell you that it worked through 100 pages, was sure with 56 of these and has some kind of clue/idea what the other 13 pages are about. The remaining 31 pages are left for the user to work on. Still, it shrunk the workload by more than half in the examples we use with customers.

WT: **Can you say something of where text analysis is going in the future?**

SK: Natural language processing (NLP) or understanding (NLU) is believed to be the next big hype within the AI. As a matter of fact, mankind has always dreamed of being able to lead a reasonable conversation with a computer. Kubrick's classic "2001" was mentally so far ahead of our society and expectations when it was first released 50 years ago (Still a must-watch from my perspective!). All of these approaches are steps in the right direction.

But recall the conversation we led over the past minutes: it is based on our understanding and not on the words we actually used in whichever sequence. The latter of which would be the approach of a classic deep learning system. We all know that the Alexa's, Siri's and Google Now's are not even close to human capabilities when it comes to language understanding—and I exclude audio processing here. My four-year-old leaves me flabbergasted daily when I see her innate capabilities when it comes to language. The next step in our point of view is combining multiple technologies and follow a more engineering-based rather than a research-based approach. That means sometimes we don't have to know why exactly something is working, it's a good start just to get it to work. Innateness, or it's lack thereof, in current approaches is the next big obstacle we need to overcome. And this is where this conversation turns philosophical. So let's stop right here. To sum it up: I'd say language is beautiful because it is hard, not in spite of.

WT: **Coming back to the coalition agreement, did your analysis cause any changes?**

ML: Not that we know of. The agreement was signed by both parties on March 12, 2018 without modifications. Germany is once more governed by a coalition of conservatives and social democrats. A coalition requires constant compromise. This can be a good way to govern. It can also fall apart.

References

[1] Marcus, G. Artificial Intelligence Is Stuck. Here's How to Move It Forward. *The New York Times*, July 29, 2017.

[2] Firth, J.R. A synopsis of linguistic theory 1930-1955. Studies in Linguistic Analysis. Oxford: Philological Society: 1–32 (1957). Reprinted in F.R. Palmer, ed. *Selected Papers of J.R. Firth 1952-1959*. Longman, London, 1968.

[3] Metz, C. Paul Allen Wants to Teach Machines Common Sense. *The New York Times*, February 28, 2018.

About the Author

Walter Tichy has been professor of Computer Science at Karlsruhe Institute of Technology (formerly University Karlsruhe), Germany, since 1986. His major interests are software engineering and parallel computing. You can read more about him at www.ipd.uka.de/Tichy.

DOI: 10.1145/3266135

