Dead Man Tweeting

David Nilsson[†], Magnus Sahlgren[⋆] and Jussi Karlgren[⋆]
[†]Nepa, Maria Skolgata 83, 118 53 Stockholm, Sweden
[⋆]Gavagai, Slussplan 9, 111 30 Stockholm, Sweden
david.nilsson@nepa.com, [firstname.lastname]@gavagai.se

Abstract

This paper presents a prototype — *Dead Man Tweeting* — of a system that learns *semantic avatars* from (dead) people's texts, and makes the avatars come alive on Twitter. The system includes a language model for generating sequences of words, a topic model for ensuring that the sequences are topically coherent, and a semantic model that ensures the avatars can be productive and generate novel sequences. The avatars are connected to Twitter and are triggered by keywords that are significant for each particular avatar.

Keywords: Chat bot, language model, natural language generation

1. Introduction

Microblog services such as Twitter attract a significant amount of non-human users. It has recently been estimated that some 8% of all Twitter users are in fact bots.[1] Many of these are spambots or simple retweeters, but there are also examples of more sophisticated and creative bots.[2] This paper introduces a new type of Twitterbot: a *semantic avatar*, which is a reactive language model built from a person's texts. The semantic avatar can react to input (in this case, tweets) that are of interest to it, and can stay updated with current events and new terminology.

The following sections describe our prototype implementation called *Dead Man Tweeting*, whose main purpose is to build semantic avatars of long-gone celebrities and to make them tweet. The current prototype features two semantic avatars: one for Russian author Fyodor Dostoyevsky (1821–1881), and one for British author and politician Winston Churchill (1874–1965).

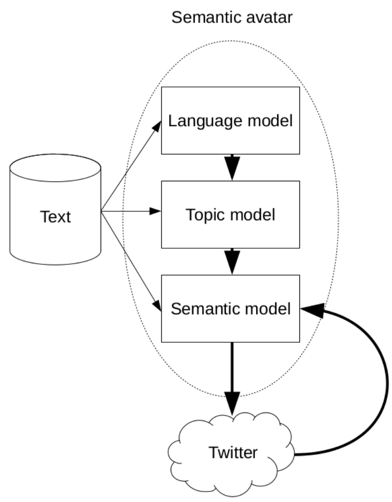

Figure 1: System architecture of Dead Man Tweeting

2. System Architecture

The semantic avatars are composed of a combination of three modules: one module for generating text sequences, another module for controlling the topical structure of each sentence, and a third module that keeps track of semantically related terms. The three modules are initialized with some given text corpus (e.g. the collected works of a dead author), and the output of the last processing step (the semantic model) is connected to Twitter. Note that the semantic module also receives input from the output domain.

2.1. The Language Model

The first module is a sequence generator based on an n-gram language model, which gives a probability distribution for a sequence $w_1, \ldots, w_m$ as $\prod_{i=1}^{m} P(w_i|w_{i-(n-1)}, \ldots, w_{i-1})$ given a context window of n words. As an example, an n-gram language model trained on a sample of general English language would give high probability to words such as "old" and "fast" when querying for the next word of the sequence "my car is very," while words such as "the" and "yogurt" would get a low probability. This fits well with our intuitions about syntactic plausibility; you seldom hear anyone say "my car is very yogurt."

The main problem with n-gram language models is that they do not generalize well, and are typically only able to reproduce sequences from the training data. More complex models such as Conditional Random Fields (CRF) (Roark et al., 2004), Recurrent Neural Networks (RNN) (Mikolov et al., 2010), and the currently very popular RNN variant Long-Short Term Memory (LSTM) (Wen et al., 2015) attempt to overcome these limitations, but they are often computationally intensive, and typically require considerable amounts of training data.

In the type of scenario we are concerned with here – building a language model based on some person's texts – we typically have access to limited amounts of training data. Even the collected works of the most productive writers are minute in comparison with the size of current standard corpora such as the British National Corpus (100 million words),[3] Wikipedia (some 1 billion words),[4] or the WaCKy

[1] qz.com/248063
[2] en.wikipedia.org/wiki/Twitterbot
[3] natcorp.ox.ac.uk
[4] dumps.wikimedia.org

corpora (more than 1 billion words).[5]

The scarcity of training data coupled with concerns of efficiency are the main reasons we opt for a simple n-gram language model with $n = 5$. We handle the limited capacity for generalization and productivity of the n-gram model by using two additional external modules: a topic model and a semantic model.

2.2. The Topic Model

The second module is a *topic model* that computes topical similarity between words. There are many ways to do this, ranging from simple methods such as the Vector Space Model (VSM) (Salton, 1964) to more advanced statistical models such as Latent Semantic Analysis (Landauer and Dumais, 1997), or probabilistic models such as Latent Dirichlet Allocation (Blei et al., 2003), and its derivatives. Again, since the more complex methods tend to be computationally intensive and require large amounts of training data, we opt in this application for the simple VSM, which represents each word as a vector $\vec{w} = [t_1, \ldots, t_n]$ where t_i is the $tfidf$-weight of the word in the ith text region.[6] Words that often occur in the same text regions get similar vectors in the VSM, and we can thus use the resulting vectors to control for topical coherence of the sequences generated by the n-gram model.

2.3. The Semantic Model

The semantic model is a *distributional semantic model* (Turney and Pantel, 2010), in which each word is represented by a *distributional vector*, $\vec{w_f} = [w_1, \ldots, w_m]$ where w_i is a function of the co-occurrence count between the focus word w_f and each context word w_i that has occurred within a window of k tokens around the focus word. Words that have co-occurred with the same *other* words (i.e. that are interchangeable in context) get similar vectors in the semantic model, which means we can use the semantic model to substitute words suggested by the n-gram model.

To implement the distributional semantic model we use the Random Indexing framework (Kanerva et al., 2000; Sahlgren et al., 2008), which accumulates distributional vectors by summing sparse random *index vectors* of fixed size that act as fingerprints for each context word. Updating a simple "one-hot" distributional model would lead to a progressively increasing dimensionality with increasing vocabulary, but the Random Indexing model accommodates the entire vocabulary in a vector space of fixed size. In effect, Random Indexing can be seen as an incremental random projection of a distributional semantic model. In this prototype, we use 2,000-dimensional vectors and 10 nonzero elements in the random index vectors. The context window is set to $k = 4$.

[5] `wacky.sslmit.unibo.it`

[6] $tfidf_{i,j} = tf_{i,j} \cdot \log \frac{N}{\mathrm{df}_i}$ where $tf_{i,j}$ is the frequency of word i in document j, df_i is the number of documents word i has occurred in, and N is the total number of documents in the data. A "document" is often a paragraph in the type of data used in these experiments.

3. The Semantic Avatars

The three modules described above are initialized with all available texts for the person in question. We have selected two prominent authors as test pilots for the semantic avatars: Fyodor Dostoyevsky, one of the authors to define the modern novel, and Winston Churchill, the 1953 Nobel Prize Laureate in Literature. Both of these authors were selected because there are substantial amounts of data freely available on sites such as Project Gutenberg.[7] For Dostoyevsky, the downloaded data contains some 1 million words, while for Churchill, the data consists of approximately 1.6 million words.

The initialization phase produces an n-gram model, a topic model, and a semantic model for each author. The n-gram model is able to generate sequences that are coherent at a very local level, but may be incoherent over longer sequences. This is where the topic model comes in. While the n-gram model learns about local structures, the topic model learns about global structures in the sense that it can recognize topically related words. Thus, if we sample the succeeding word given $n - 1$ previous words using the n-gram model, we can use the topic model to filter out words that are not topically related to the words that are already in the current sequence. This topic filter ensures that each sequence is topically coherent even over longer sequences.

However, we are still only able to reproduce short sequences of words we have already seen in the original data, and since our chosen subjects have not generated any words for a long time, their avatars have quite outdated vocabulary. Enter the semantic model, which is partly trained on the subject's texts, but which also receives an input stream from relevant news channels. In the case of Dostoyevsky, we use feeds from Reuters and Russia Today, and in the case of Churchill we use a feed from BBC news. These input streams are continuously fed into the semantic model, which thereby accrues an expanding vocabulary that may contain terms that were never used by the avatar's ancestor. The semantic model thus constitutes an avatar's ability to learn new words that are in some sense related to what it already knows. The avatars may use the semantic model to randomly substitute words generated by the n-gram model to other words that are highly related in the semantic model. The effect is that the avatars are able to use terminology that was not present in the original data, which means they can now be productive and generate completely novel sequences.

The combination of the three different modules enables the semantic avatars to not only generate topically coherent sequences, but also to be productive and utter new things. However, we also want the avatars to be reactive and to interact with the world. We therefore connect each avatar to Twitter and add some accounts to follow. Each avatar is triggered by tweets containing words of particular interest to the avatar, which are extracted from the training data using $tfidf$. If the avatar is triggered by a tweet containing a keyword, it formulates a reply containing that keyword, and posts it on Twitter.

[7] `gutenberg.org`

4. Examples

Figures 2 and 3 give examples of how Churchill's semantic avatar (@churchillDMT) and Dostoyevsky's semantic avatar (@dostoyevskyDMT) interact with other Twitter users; in the first example, Churchill's semantic avatar refutes a claim made by the Churchill Center, and in the second example, Dostoyevsky's semantic avatar comments on a tweet by the Russian Ministry of Foreign Affairs. The two avatars sometimes also comment on each other's tweets, as in Figure 4 when Churchill's semantic avatar comments on a tweet by Dostoyevsky's avatar (which in its turn is a reply to another tweet). There are also examples of when other Twitter users reply to the avatar's tweets, apparently considering them (correctly so) to be communicative users, as in Figure 5.

Figure 2: Churchill's semantic avatar answers the Churchill Center.

Figure 3: Dostoyevsky's semantic avatar answers the Russian Ministry of Foreign Affairs.

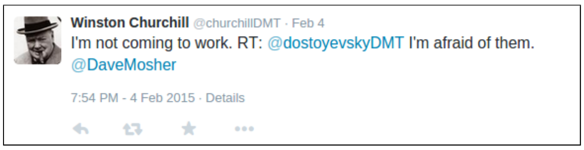

Figure 4: Churchill's and Dostoyevsky's semantic avatars having a conversation.

5. Discussion

The examples above are of course hand-picked to demonstrate when the avatars produce intelligible (or at least entertaining) conversation. Admittedly, the avatars do not always produce meaningful output. Two telling examples are @dostoyevskyDMT's tweet "`But she after three years later!`" and @chuchillDMT's tweet "`But most men in the working women relief.`"

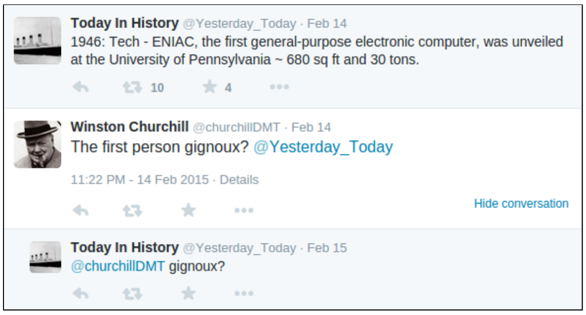

Figure 5: Churchill's semantic avatar converses Today In History, but makes little sense.

There are many reasons for the varied quality of the avatar's responses. First of all, there are a number of parameters in the interaction between the various modules that can be tuned and optimized; when optimizing for short sequences, the avatars produce very terse responses, while when optimizing for longer sequences, the avatars risk becoming incoherent. Furthermore, there is currently no topic correction between an avatar's generated text and the tweet it reacted upon. We are at this stage merely requiring the presence of the trigger keyword, which is obviously a too lenient requirement that can cause incoherent responses. A component which would hold to a topic-comment structure, with a topic selected from the input stimulus but with a freer progression towards associated topics would give some level of discourse backbone to the text.

Also, the semantic model may cause alterations that make the sequence less intelligible, which could potentially be solved by letting the topic model override semantic expansions in order to preserve topical coherence.

It is of course well-known that language generation for open domain text is a very difficult task. Most previous studies and applications of natural language generation has focused on closed domains, such as weather forecasts (Belz, 2008), health care (Harris, 2008), or review summarization (Di Fabbrizio et al., 2013), and commercial vendors such as Automated Insights[8], Narrative Science[9], and Yseop[10], also tend to focus on domain-specific solutions.

One of the reasons for choosing Twitter as the medium for the semantic avatars rather than, e.g., email or blogs is that the length restriction of tweets makes it easier to generate open domain text and that the rhapsodic nature of the generated discourse better fits expected human behaviour. However, the strength of the Twitter medium – its brevity – is also a weakness in the current application, since it is difficult to replicate an author's stylistic and topical peculiarities, and to demonstrate semantic productivity, in such short sequences. As a consequence, many of the avatars' tweets take the form of questions, exclamations, or otherwise quite terse statements, where it is difficult to discern any personality traits.

[8] `automatedinsights.com`
[9] `narrativescience.com`
[10] `yseop.com`

Even so, the dialogic nature of Twitter facilitates a benevolent interpretation of the avatar's utterances. Even very short and repetitive tweets, such as @dostoyevskyDMT's "`yes yes yes`" may be interpreted as a perfectly viable reply given the context "`Apple's App Store says no to guns but yes to weed`" tweeted by @TechCrunch. A Turing test based on Twitter would probably be passable using technologies such as those presented here, which would most likely not be the case given other communication channels.

We of course make no claims about any communicative intelligence behind the semantic avatars; they simply put words together – but they do put words together under much the same type of constraints as is characteristic of human communicative competence: syntactic (which we simplify to sequential) consistency, topical coherence, and semantic productivity. At the current state of the semantic avatars, the sequences they generate make sense on Twitter about as often as they seem to be gibberish. More data would likely improve the quality of the avatars.

6. Texts and Resources

We have implemented the semantic avatars not to need any other resources than text from the avatar's living correlates: no need for grammars, hand-encoded knowledge structures, or other expensive conceptual models. Access to a sizeable text collection is thus a necessary prerequisite for launching an avatar. The model, as implemented in these present experiments, relies on both the collection of background text to give the avatars their personal touch, as well as access to text on current events to allow them to evolve with current language usage, and to provide a general language background. Collections of general language background data from various historical periods and covering various types of domains and styles would be a very useful resource for the creation of historic avatars.

The various modules of the avatars can be seen as individual resources for further development of personalized chat bots. Harvesting sequence, topical, and semantic modules from a large number of subjects would enable the creation of *modular* avatars, where it would be possible to select a sequence model from one subject, a topical model from another, and a semantic model from a third. The resulting combined avatar would have the syntactic qualities from one person, the topical knowledge of another, and the semantics of a third; imagine a bot that would have the syntactic qualities of James Joyce, the topical knowledge of Aristotle, and the semantics of Nietzsche.

7. Directions of Application

We have used the prototype to revive two prominent dead authors, and one could imagine applications geared towards *mind uploading*[11] – i.e. the postmortem preservation of personality – which would enable people to converse with the dead, and to solicit opinions and statements on current events. A bit less science fiction-like, one might also use this type of system for creating communicative avatars for entertainment purposes (e.g. chat bots, or in-game characters), or for creating aggregated views on entire text collections, such as a political party's or a company's internal documents. Currently we are including a Dead Man Tweeting component as a fall-back resource in a chat bot geared towards school children for a specific field of discussion. When the conversation veers off target or leaves the fairly limited area of competence the bot has been equipped with (which can be expected when conversing with teenagers), the chat bot uses a Dead Man Tweeting component to generate a fall-back response. In this case, the text corpus used for training is a parameter to be experimented with to get the most natural and entertaining output.

8. Conclusion

This paper has introduced the notion of a semantic avatar, and has provided examples of output from two different avatars: @dostoyevskyDMT and @churchillDMT. The avatars are based on a combination of a language model that generates sequences of words, a topic model that ensures the generated sequences are topically coherent, and a semantic model that equips the avatars with semantic productivity.

Despite leaving room for improvement, we find that the examples produced by the Dead Man Tweeting prototype demonstrate the potential for semantic avatars as a way to build autonomous reactive systems that can interact linguistically with its surroundings.

9. References

Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, October.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Di Fabbrizio, G., Stent, A., and Gaizauskas, R. (2013). Summarizing opinion-related information for mobile devices. In Amy Neustein et al., editors, *Mobile Speech and Advanced Natural Language Solutions*, pages 289–317. Springer New York.

Harris, M. D. (2008). Building a large-scale commercial NLG system for an EMR. In *Proceedings of INLG*, pages 157–160.

Kanerva, P., Kristofersson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of CogSci*, page 1036.

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048.

Roark, B., Saraclar, M., Collins, M., and Johnson, M. (2004). Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of ACL*.

[11] `en.wikipedia.org/wiki/Mind_uploading`

Sahlgren, M., Holst, A., and Kanerva, P. (2008). Permutations as a means to encode order in word space. In *Proceedings of CogSci*, pages 1300–1305.

Salton, G. (1964). A flexible automatic system for the organization, storage, and retrieval of language data (SMART). In *Report ISR-5 to NSF*, Cambridge, MA. Harvard Computation Laboratory.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, January.

Wen, T.-H., Gasic, M., Mrksic, N., Hao Su, P., Vandyke, D., and Young, S. J. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of EMNLP*, pages 1711–1721.