A Turing Test to Evaluate a Complex Summarization Task

Alejandro Molina[1], Eric SanJuan[1,2], and Juan-Manuel Torres-Moreno[1,2,3]

[1] LIA, Université d'Avignon et des Pays de Vaucluse,
339 chemin des Meinajaries, Agroparc BP 1228, F-84911 Avignon Cedex 9, France
alejandro.molina-villegas@alumni.univ-avignon.fr,
{eric.sanjuan,juan-manuel.torres}@univ-avignon.fr
[2] Brain & Language Research Institute,
5 avenue Pasteur, 13604 Aix-en-Provence Cedex 1, France
[3] École Polytechnique de Montréal,
2900 Bd Edouard-Montpetit Montréal, QC H3T1J4, Canada

Abstract. This paper deals with a new strategy to evaluate a Natural Language Processing (NLP) complex task using the Turing test. Automatic summarization based on sentence compression requires to asses informativeness and modify inner sentence structures. This is much more intrinsically related with real rephrasing than plain sentence extraction and ranking paradigm so new evaluation methods are needed. We propose a novel imitation game to evaluate Automatic Summarization by Compression (ASC). Rationale of this Turing-like evaluation could be applied to many other NLP complex tasks like Machine translation or Text Generation. We show that a state of the art ASC system can pass such a test and simulate a human summary in 60% of the cases.

1 Introduction

Alan Turing predicted that computers will be better at playing complex board games like chess than to chat with humans in an open world. Natural Language Processing (NLP) appeared in 1951 to be one of the greatest challenges for computers. Surprisingly, some tasks like automatic summarization appeared to be easier than anticipated when considering extracts instead of abstracts [1]. Summarization by extraction often consists in segmenting the text to be summarized into sentences and to apply scoring methods to rank sentences by decreasing informativity. In this simplified task, resulting short summaries are often readable because they use real sentences. The main difficulty when dealing with longer summaries involving ten or more sentences is to avoid breaking anaphora. This is handled using simple heuristics like displaying top ranked sentences in the order they appear in the original text. Since local text grammaticality is ensured by keeping entire sentences, resulting summaries often give the illusion that they were written by a human. Moreover, under the assumption that the produced summary is readable, summary informativeness can be evaluated using measures like ROUGE given on a set of reference summaries or Jensen-Shannon/Kullback-Leibler metrics if no reference summary is available [2–4].

P. Forner et al. (Eds.): CLEF 2013, LNCS 8138, pp. 75–80, 2013.

The task becomes much more complex if computer cut and compress sentences like humans do since this implies the ability to understand and modify inner sentence structures. Discourse structure among other implicit semantic relations play a key role [5]. Moreover there are usually several correct ways to compress a sentence and human experts often disagree on which is the best one. When trying to build a reference corpus of compressed sentences, inter agreement between annotators is low, even to decide if a sentence should be shortened in the summary or not. Automatic Summarization by Compression (ASC) requires to handle a high level of incertainty in the decision process since there is not a best way to compress a sentence, only observations that sometimes humans prefer one way rather than another one [6]. Not only the task itself is difficult but it cannot be evaluated using existing methods. Using sentence compression to produce a summary not always improve informativeness scores and can produce unreadable summaries. Therefore, actual state of the art evaluation metrics for automatic summarization discourage thorough investigations if a computer can handle or not ASC.

In this paper we show that coming back to the original idea of a Turing test, it is possible to set up a simple imitation game to evaluate ASC. We also show that a state of the art system that learns human behavior using simple regression analysis [6] can pass this test on short summaries and give the illusion to human referee that the summary was written by a human. Moreover this imitation game is clearly adapted to crowd sourcing through Internet and can be used to evaluate large amount of systems at a reasonable cost.

The rest of the paper is organized as follows. Section 2 goes back to the general definition of a Turing test. Section 3 details the imitation game that we propose to evaluate ASC in a pragmatic way. Section 4 shows statistical evidence that a state of art ASC system can pass the test. Finally, section 5 opens perspectives on how this evaluation methodology can contribute to the improvement of effective ASC systems.

2 Back to Turing Test

As suggested by Alan Turing, a test to evaluate the ability of a computer to handle a human mind task should involve:

- an interaction with humans where the computer tries to give the illusion that it is human,
- a clear evaluation metric that allows the reproducibility of the experiment,
- a gateway to the open world to explore beyond restricted contexts and closed world assumptions.

Our main motivation relies on the fact that, to the best of our knowledge, there is no summarization evaluation methodology that encourages research on advanced NLP tasks like summarization by sentence compression. We therefore suggest to come back to Turing's initial motivations[7] when imaging imitation games to answer the controversial philosophical question "do computers have a mind? " without having to define what "mind" means. The question then becomes "what are the common human intellectual tasks that a computer can handle? " These are the roots of theoretical computer science where tasks almost useless for technical applications can be fundamental to understand computers' real limits. ASC can have many applications in our interconnected world but we claim that its main interest relies on the theoretical study of computer capabilities.

In a the original imitation game defined by Turing in [7], there are two players and one assessor. The first player is a human (A) and the second a computer (B). Another human (C) plays the role of the assessor and has to guess the real nature (human or computer) of the two other players. The assessor cannot see the other players, he can just interact with them through a more or less restricted interface that at least allows to exchange written messages. The assessor asks questions through the interface and has to distinguish between answers given by the human player and those sent by the computer.

Turing imagined advanced imitation games to study the spectrum of Artificial Intelligence and compare it to the human mind. However, as pointed out by [8], Turing entrusted interaction through natural language. In our case, we intend to study the method of interacting itself related to NLP and its linguistic functionalities based on summary generation. Indeed, in the general case of a Turing test, the assessor is not allowed "to see" the players. This is to ensure that he focus on functional aspects and not on appearances. It then seems natural to adapt the imitation game to NLP tasks that try to reproduce human ability to handle texts like summarization. We do not consider tasks that cannot be carried out without computer assistance like Information Retrieval from large collections. Only intellectual tasks that can easily be accomplished by non experts meanwhile there are real challenges for an automatic system.

3 Imitation Game to Evaluate ASC

We consider the following imitation game involving a human player (A), a computer (B) and a human assessor (C). A and B are asked to write one summary for each of some texts. After some time, an interface between C and the players dispatches the summaries at random, just checking that each player have the same number of texts. So C does not know who between A and B wrote each summary and has to guess the correct author for each text.

This setting follows Turing's idea of an interactive game between two humans and a computer. However, one difficulty to carry it out is that humans need time to write a summary meanwhile it is necessary to reproduce the same experiment at least 30 times to expect some statistical evidence if there is a regular winner between A and B. To adapt this game to standard crowd-sourcing evaluations, we decided to consider a team of $n \geq 30$ extra assessors $(C_1, \cdots, C_n)$, a team of different human players $(A_1, \cdots, A_k)$ and a set of different computational strategies or systems $(B_1, \cdots, B_m)$. The main drawback of this adaptation is the lack of real interactivity. The main advantage is that this rationale could be adapted to many other domains using machine learning to simulate particular human brain functionality (for instance, NLP complex tasks).

Let us give some details about the way we implemented this game to evaluate ASC systems. 60 post-graduated students accepted to participated in this simulation game, 6 of them were asked to write summaries $(A_1, \cdots, A_6)$ and the 54 other participated as assessors $(C_1, ..., C_{54})$. It must be note that all of them, team (A) and team (C), expected team (B), the systems, to fail. 12 texts were selected from the RST Spanish Tree

Bank[9] at random. Summaries of these texts have been written down by team (A). We chose the ASC systems derived from [6] as the team (B). These summarizers are based on machine learning techniques that emulate the way annotators agree or not with a sentence compression using two discourse segmentation strategies: DiSeg [10][1] (B_1) and CoSeg (B_2). It has been shown in [5] that humans tend to remove complete discourse units from sentences when they try to compress them. As anticipated for a so subjective task, inter agreement between assessors was very low but enough to carry out a regression analysis and learn to predict the probability of a particular sentence compression to be accepted by humans. Three summaries of different length (short, medium and long) were generated using DiSeg (B_1), and three other ones also of different length were generated using CoSeg (B_2). All assessors read the 12 summaries and for each they tried to guess if the author of the summary was a human or a computer. They did not know that exactly half of the summaries were automatically generated.

4 Results

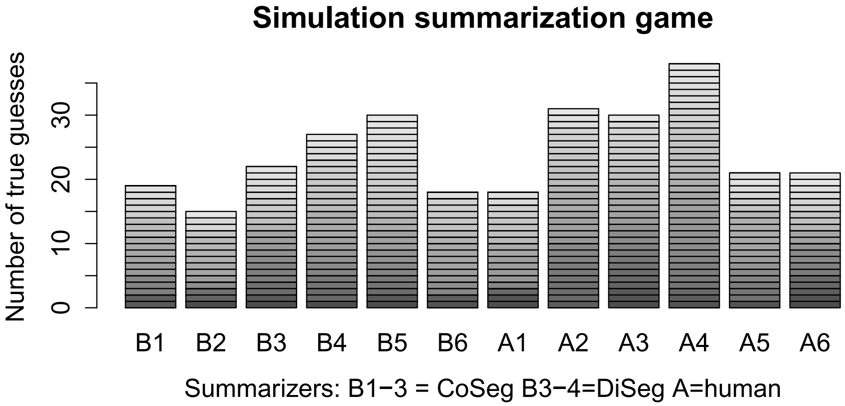

Fig. 1. summarization simulation game: each bar shows the number of correct guesses (Human or Computer) for each summarizer

Figure 1 shows the results of the simulation game. where numbers next to (B) team represent the three different lengths (1:short, 2:medium, 3:expanded). In this figure, bars for summaries written by team (A) are expected to be higher if they are good quality summaries meanwhile bars for team (B) are expected to be low since they intent to mislead the assessor. It appears that over the six authors of summaries, only three manage to write summaries that more that 60% of the assessors think they can not be automatically generated. Meanwhile, player (B_1), the automatic system DiSeg, manage to mislead the assessors on 60% of long summaries and player (B_2), CoSeg system on short and medium summaries.

[1] http://diseg.termwatch.es

Plot 2 shows the median normalized frequency of times that an assessor thinks the summary has been written by a human. The first boxplot shows it over the twelve summaries each assessor has to read. The second and third boxplots over the three summaries generated using CoSeg and DiSeg respectively. The fourth is over the six summaries by humans and the last one is restricted to the three best authors A_2, A_3 and A_4. These boxplots suggest that summary quality by three best authors (last boxplot) is above average among summaries written by real authors (fourth boxplot) and among overall summaries (first boxplot). However, according to a Wilcoxon test with a p-value lower than 0.01, only the differences between best human summaries and all human summaries is statistically significant. The difference between best human summaries and overall summaries is not. Similarly, CoSeg summaries outperform DiSeg summaries since the median frequency it misleads assessors is significantly higher (p-value < 0.05) meanwhile all other differences are not statistically significant. In particular there is not statistical evidence based on Wilcoxon rank sum test with continuity correction that an assessor thinks that the summary has been done by a human author when reading a summary generated by one of the automatic summarizers tested here, than one really done by a human author.

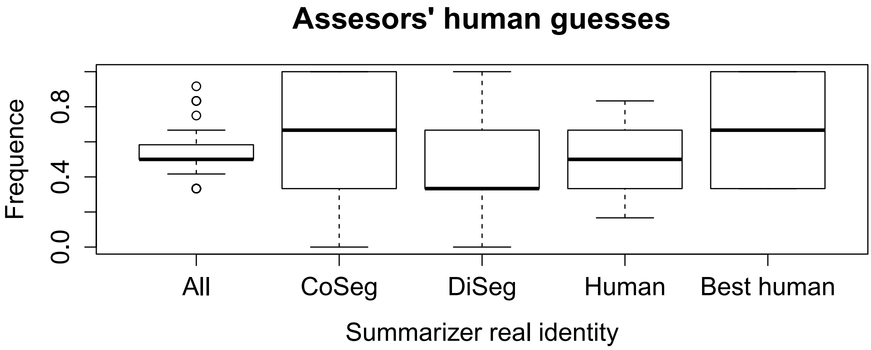

Fig. 2. Boxplots showing the median number of times that an assessor thought it was a summary produced bay a human for each set of six summaries and each subset of three summaries automatic/human

5 Discussion

We have use a Turing test to evaluate two state of the art automatic summarizers where usual evaluation protocols failed to differentiate between quality levels among the two system outputs. The principal argument is that if human and machine productions could not be differentiated, then they might have similar quality.

The experiment set up here with 60 human players gives statistical evidence that one system outperforms the other. But we also find out that human juges cannot differentiate between written by an author abstracts and automatically generated summaries when using sophisticated methods as ASC that goes beyond sentence extraction and ranking. Results are promising, though this to be checked out by setting up a larger crowdsourcing experiment and testing some enhancements. For instance, this first experiment cannot quantify the gap in quality between the good and bad summaries. However, mixing human and machine outputs using Turing test adapted to specific tasks could represent a new evaluation paradigm that need to be more explored. Even more, we think that it could have broader applications.

Acknowledgments. We would like to thank our contributors for their help with the corpus annotation, specially to Gerardo Sierra and the Linguistics Engineering Group UNAM. This work was partially supported by the CONACyT grant 211963 and the ANR Imagweb project (ANR-12-CORD-0002).

References

1. Tratz, S., Hovy, E.: Summarisation Evaluation Using Transformed Basic Elements. In: Workshop Text Analysis Conference (TAC 2008), Gaithersburg, MD, USA (2008)
2. Louis, A., Nenkova, A.: Automatically Evaluating Content Selection in Summarization without Human Models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapour, August 6-7, pp. 306–314. ACL (2009)
3. Saggion, H., Torres-Moreno, J.-M., da Cunha, I., SanJuan, E.: Multilingual summarization evaluation without human models. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING 2010), Beijing, Chine, pp. 1059–1067. ACL (2010)
4. Torres-Moreno, J.-M., Saggion, H., da Cunha, I., SanJuan, E.: Summary Evaluation With and Without References. Polibits: Research Journal on Computer Science and Computer Engineering with Applications 42, 13–19 (2010)
5. Molina, A., Torres-Moreno, J.-M., SanJuan, E., da Cunha, I., Sierra, G., Velázquez-Morales, P.: Discourse segmentation for sentence compression. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part I. LNCS, vol. 7094, pp. 316–327. Springer, Heidelberg (2011)
6. Molina, A., Torres-Moreno, J.-M., SanJuan, E., da Cunha, I., Martínez, G.E.S.: Discursive sentence compression. In: Gelbukh, A. (ed.) CICLing 2013, Part II. LNCS, vol. 7817, pp. 394–407. Springer, Heidelberg (2013)
7. Turing, A.M.: Computing machinery and intelligence. Mind 59(236), 433–460 (1950)
8. Harnad, S.: Minds, Machines and Turing. Journal of Logic, Language and Information 9(4), 425–445 (2000)
9. da Cunha, I., Torres-Moreno, J.-M., Sierra, G.: On the Development of the RST Spanish Treebank. In: Linguistic Annotation Workshop, pp. 1–10. The Association for Computer Linguistics (2011)
10. da Cunha, I., SanJuan, E., Torres-Moreno, J.-M., Lloberes, M., Castellón, I.: DiSeg 1.0: The first system for Spanish Discourse Segmentation. Expert Systems with Applications 39(2), 1671–1678 (2012)