# Automated Model of Comprehension V2.0

Dragos-Georgian Corlatescu[1], Mihai Dascalu[1,2(✉)], and Danielle S. McNamara[3]

[1] University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania
{dragos.corlatescu,mihai.dascalu}@upb.ro
[2] Academy of Romanian Scientists, Street Ilfov, Nr. 3, 050044 Bucharest, Romania
[3] Department of Psychology, Arizona State University, PO Box 871104, Tempe, AZ 85287, USA
dsmcnama@asu.edu

**Abstract.** Reading comprehension is key to knowledge acquisition and to reinforcing memory for previous information. While reading, a mental representation is constructed in the reader's mind. The mental model comprises the words in the text, the relations between the words, and inferences linking to concepts in prior knowledge. The automated model of comprehension (AMoC) simulates the construction of readers' mental representations of text by building syntactic and semantic relations between words, coupled with inferences of related concepts that rely on various automated semantic models. This paper introduces the second version of AMoC that builds upon the initial model with a revised processing pipeline in Python leveraging state-of-the-art NLP models, additional heuristics for improved representations, as well as a new radiant graph visualization of the comprehension model.

**Keywords:** Comprehension model · Natural language processing · Semantic links · Lexical dependencies

## 1 Introduction

Comprehension is fundamental to learning. While there is much more to learning (e.g., discussion, project building, problem solving), understanding text and discourse represents a key starting point when attempting to learn or relearn information. How well a reader understands text or discourse depends on many factors, including individual differences such as reading skill, prior knowledge of the domain or world, motivation, and goals. Comprehension also depends on the nature of the text – the difficulties imposed by the words in the text, the complexity of the syntax, and the flow of the ideas, or cohesion.

Cohesion between ideas can emerge from overlap between explicit words (e.g., nouns, verbs), implied words (anaphor), semantically related words, semantically related ideas, and the underlying parts of speech (i.e., parts of speech, syntactic overlap). When there is greater overlap, text is easier to understand. Cohesion gaps, by contrast, require inferences to make connections between the ideas. If the reader has little knowledge of the domain or the world, low cohesion text impedes comprehension [1]. For example,


I. Roll et al. (Eds.): AIED 2021, LNAI 12749, pp. 119–123, 2021.
https://doi.org/10.1007/978-3-030-78270-2_21

if the text is too complex, readers may struggle to understand it or event abandon the process; and on the other side, if too simple, readers may quickly lose focus or interest. Thus, designing reading materials suited for learners is an important aspect for educators as well as writers when targeting a specific audience.

The automated model of comprehension (AMoC) simulates the mental representation constructed by hypothetical readers, by building syntactic and semantic relations between words, coupled with inferences of related concepts that rely on various semantic models. AMoC offers the user the ability to model various aspects of the reader by modifying various parameters related to readers' knowledge, reading skill, and motivation (i.e., activation threshold, maximum active number of concepts per sentence, maximum number of semantically related concepts, and the type of knowledge model). This paper introduces an updated version of the automated model of comprehension (AMoC version 2.0), which is freely available online at http://readerbench.com/demo/amoc.

AMoC builds on the Construction Integration (CI) model [2], which introduced a semi-automated cyclical process to simulate reading, as well as the Landscape Model [3], which inherited the ideas from the CI model and provided a visual representation of the activation scores belonging to the concepts in the text. We describe a revised version of AMoC that provides several enhancements: a) an improved processing pipeline rewritten in Python, b) additional heuristics introduced to better model human constructs, and c) a radiant graph visualization to highlight the model's capabilities.

## 2 Method

The codebase for AMoC version 2 is developed in Python, rather than Java. This decision was influenced by the progress and the interest of the artificial intelligence community into libraries written in this programming language such as Tensorflow [4] and Pytorch [5], that are frequently used in general neural network projects, and SpaCy [6], which is an open source tool for NLP tasks. Additionally, the *ReaderBench* framework [7], previously implemented in Java and used in first version of AMoC, migrated to Python, offering improved functionalities based on state-of-the-art NLP models.

AMoC uses three customizable parameters: *minimum activation score* (the activation score required by a word to be active in the mental model), *maximum active concepts* (the maximum number of words that can be retrieved in the mental model) and *maximum dictionary expansion* (the number of words that can be inferred each sentence). Those three parameters and the target text are processed by the model. The processing begins by automatically splitting the text into sentences using ReaderBench. For the current study, *ReaderBench* Python uses SpaCy to split and store the relations between words. Next, the syntactic graph for each sentence is computed and stored in the model's memory. The difference between the AMoC v1 and v2 is that the coreferences are obtained and replaced using NeuralCoref [8] in the later version, while in the older version a Stanford Core NLP [9] module was applied; Wolf [10] argues that NeuralCoref obtains better overall performance. Additionally, the syntactic parser from SpaCy performs slightly better than Stanford CoreNLP [11, 12]: SpaCy UAS 92.07, LAS 90.27 versus Stanford CoreNLP UAS 92.0, LAS 89.7. The process includes:

1. Each sentence is processed iteratively and each content word (noun or verb) in the sentence is added to the graph if it was not present before, or its activation score is incremented by 1 if the reader has previously encountered the concept in the text.
2. When processing a sentence, the top 5 similar concepts are inferred using WordNet (to extract synonyms and hypernyms) and word2vec [13]. The word2vec models trained on TASA [14], COCA [15], or Google News [13] are considered to reflect different levels of reading proficiency in terms of exposure to language
3. The inferred words from a sentence are filtered based on two criteria: they must have a semantic similarity with the sentence of at least .30 (a value argued by Ratner [16]) and they must have a Kuperman Age of Aquisition [17] score $< 9$ (i.e., the word is accessible to an average reader).
4. Finally, all of the semantic links in the graph are removed, and the semantic nodes are sorted based on the similarity with the current sentence. Then, only the top *maximum dictionary expansion* concepts are added to the graph.

The key differences between the two versions of AMoC are in the second and third steps. The first version of AMoC used only the synonyms extracted from WordNet; the current version also uses the hypernyms and words extracted from a word2vec language model. Also, the filtering process in the older AMoC version did not include the Age of Acquisition score to represent the potential difficulty of the words.

After the semantic nodes are added to the graph, a modified PageRank algorithm [18] is run in order to spread activation between concepts and then a normalization step is applied. Lastly, after all these operations, nodes become or remain active if they have a score above the *minimum activation score*; otherwise, they are deactivated.

## 3 Results

A demo of AMoC v2 is available on the ReaderBench website with varying parameters. Since the release of the first model the UI was updated with a highly customizable radiant graph. Figure 1 illustrates the last sentence from the "Knight" story used to showcase the Landscape Model – http://www.brainandeducationlab.nl/downloads; the caption uses TASA as semantic model, a minimum activation threshold of .30, 20 maximum active concepts per sentence, and 2 maximum semantically related concepts introduced for each word in the original text. The inner circle depicts in blue text-based information that is still active, while the outer circle contains semantically inferred concepts in red and grayed out inactive concepts. When hovering over a node, the corresponding edges are colored, and the related concepts are marked in bold. While considering text-based information, "princess" is related to "dragon", "armor", "marry", and "knight", whereas from a semantic perspective, "princess" is linked to "damsel", "prince", and "sword"; all concepts and underlying links are adequate concepts for the selected story.

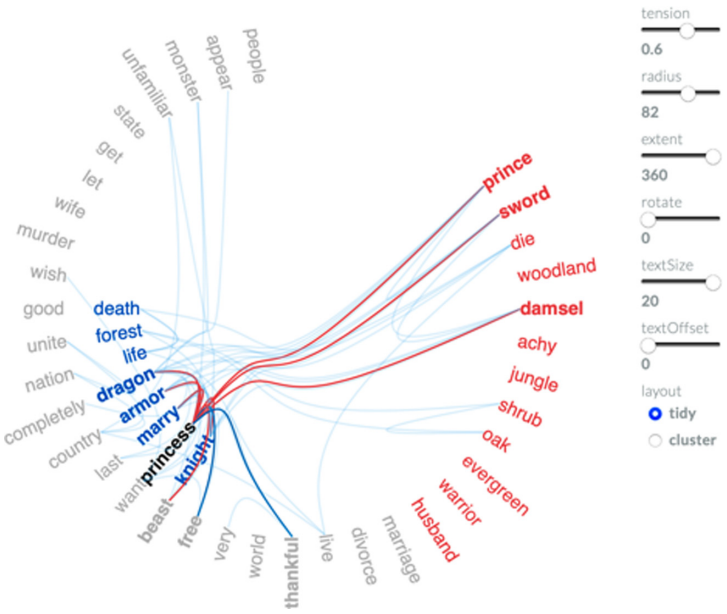


**Fig. 1.** AMoCv2 Radiant graph visualization of the last sentence from the Knight story.

## 4 Conclusions and Future Work

AMoC provides a fully automated means to model comprehension by leveraging current techniques in the Natural Language Processing field. The second version of AMoC described in this research provides an improved method and optimizations at the code base level in comparison to its predecessor, combined with a more rapid execution time. Additionally, a new and highly customizable method for concept graph visualization was added to the ReaderBench website.

In future research, we will further test the predictiveness of AMoC by applying it to previous studies that examined text comprehension. From a more technical perspective, we intend to evaluate potential advantages of using BERT contextualized embeddings [19], rather than word2vec. Our overarching objective is to comprehensively account for word senses and their contexts within sentences, paragraphs, texts, and language.

**Acknowledgments.** This research was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number TE 70 *PN-III-P1-1.1-TE-2019-2209,* ATES – "Automated Text Evaluation and Simplification", the Institute of Education Sciences (R305A180144 and R305A180261), and the Office of Naval Research (N00014-17-1-2300; N00014-20-1-2623). The opinions expressed are those of the authors and do not represent views of the IES or ONR.

## References


1. McNamara, D.S., Kintsch, E., Songer, N.B., Kintsch, W.: Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. Cogn. Instr. **14**(1), 1–43 (1996)
2. Kintsch, W., Welsch, D.M.: The construction-integration model: a framework for studying memory for text. In: Relating Theory and Data: Essays on Human Memory in Honor of Bennet B. Murdock., pp. 367–385. Lawrence Erlbaum Associates, Inc., Hillsdale (1991)
3. Van den Broek, P., Young, M., Tzeng, Y., Linderholm, T.: The Landscape Model of Reading: Inferences and the Online Construction of a Memory Representation. The Construction of Mental Representations during Reading, pp. 71–98 (1999)
4. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283. {USENIX} Association, Savannah, GA, USA (2016)
5. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8026–8037, Vancouver, BC, Canada (2019)
6. Honnibal, M., Montani, I.: Spacy 2: natural language understanding with bloom embeddings. In: Convolutional Neural Networks and Incremental Parsing, vol. 7, no. 1 (2017)
7. Dascalu, M., Dessus, P., Trausan-Matu, Ş, Bianco, M., Nardy, A.: ReaderBench, an environment for analyzing text complexity and reading strategies. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS (LNAI), vol. 7926, pp. 379–388. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_39
8. huggingface: Neuralcoref, Accessed 30 Dec 2020. https://github.com/huggingface/neuralcoref (2020)
9. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for computational Linguistics: System Demonstrations, pp. 55–60. The Association for Computer Linguistics, Baltimore, MD, USA (2014)
10. Wolf, T.: State-of-the-art neural coreference resolution for chatbots, Accessed 30 Dec 2020. https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30 (2017)
11. Explosion: SpaCy, Accessed 9 Feb 2021. https://spacy.io/usage/facts-figures#benchmarks (2016–2021).
12. The Stanford Natural Language Processing Group: Neural Network Dependency Parser, Accessed 9 Feb 2021. https://nlp.stanford.edu/software/nndep.html
13. Google: word2vec, Accessed 30 Nov 2020. https://code.google.com/archive/p/word2vec/ (2013)
14. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
15. Davies, M.: The corpus of contemporary American English as the first reliable monitor corpus of English. Literary Linguist. Comput. **25**(4), 447–464 (2010)
16. Ratner, B.: The correlation coefficient: its values range between+1/−1, or do they? J. Target. Meas. Anal. Mark. **17**(2), 139–142 (2009)
17. Kuperman, V., Stadthagen-Gonzalez, H., Brysbaert, M.: Age-of-acquisition ratings for 30,000 English words. Behav. Res. Methods **44**(4), 978–990 (2012)
18. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. **30**(1–7), 107–117 (1998)
19. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)