Semi-Automatic Generation of Intelligent Curricula to Facilitate Learning Analytics

Angel Fiallos
Escuela Superior Politecnica del Litoral
Guayaquil, Ecuador
anfiallos@fiec.espol.edu.ec

Xavier Ochoa
New York University
New York, New York
xavier.ochoa@nyu.edu

ABSTRACT

Several Learning Analytics applications are limited by the cost of generating a computer understandable description of the course domain, what is called an Intelligent Curriculum. The following work contributes a novel approach to (semi-)automatically generate Intelligent Curriculum through ontologies extracted from existing learning materials such as digital books or web content. Through a series of natural language processing steps, the semi-structured information present in existing content is transformed into a concept-graph. This work also evaluates the proposed methodology by applying it to learning content for two different courses and measuring the quality of the extracted ontologies against manually generated ones. The results obtained suggest that the technique can be readily used to provide domain information to other Learning Analytics tools.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; • **Applied computing** → **Computer-assisted instruction**;

KEYWORDS

intelligent curriculum, ontologies, NLP

ACM Reference Format:
Angel Fiallos and Xavier Ochoa. 2019. Semi-Automatic Generation of Intelligent Curricula to Facilitate Learning Analytics. In *The 9th International Learning Analytics Knowledge Conference (LAK19), March 4–8, 2019, Tempe, AZ, USA.* ACM, New York, NY, USA, Article 4, 5 pages. https://doi.org/10.1145/3303772.3303834

1 INTRODUCTION

Intelligent Curriculum has been identified since the inception of Learning Analytics [1] [2] as one of the enablers of a data-informed decision support systems in education. An Intelligent Curriculum can be defined as the representation of the knowledge domain usually taught in a course in a way that is amenable to be understood and processed by a computational system. The most common representation that fulfill this requirement is an ontology [3]. Once the curriculum is represented as an ontology, several existing Learning Analytics applications could use this information to recommend learning materials [4], to automatically sequence learning activities [5], to evaluate the quality of contributions in online forums [6] or to provide visual feedback to students about their progress [7], among others.

As useful as having an Intelligent Curriculum could be, the cost of its manually create these course-oriented ontologies is high [8]. Domain experts are rarely experts in semantic technologies and vice versa. Moreover, the cost of maintaining these ontologies up-to-date to the natural changes of the courses and important topics is not trivial [9]. It could be argued that the cost of creation and maintenance of small-scale ontologies has limited their use in the field of Learning Analytics.

This lack of course ontologies, however, is opposed by the abundance of semi-structured information in the form of learning materials. For most disciplines, it is easy to find learning materials created by domain experts that contain not only relevant content, but also structure (table of contents, links, sections, references) that make explicit the semantic relation between different topics and subtopics. This work proposes and tests the idea of using existing learning resources such as digital books, web-based tutorials or existing syllabus text in digital format as sources to (semi-)automatically build and maintain course-based ontologies that could realize the concept of Intelligent Curriculum.

This article is structured as follow: Section 2 presents a brief discussion of related work and how this work uniquely contribute to the state-of-the-art. Section 3 describes in detail the inner workings of the automatic extraction of course-centric ontology. In Section 4, the described system is applied to two different domains and the results are presented and evaluated. The article ends with conclusions and possible inmediate Learning Analytics applications.

2 RELATED WORK

Automatically extracting ontologies from existing semi-structured text is not a novel idea. Wong et al. [10] present a survey of several techniques to do it. In the educational field, the most successful approach has been the one proposed by Guerra at al. [11] to use topic modelling to create links between textbooks chapters and subchapters. This network of content was then used to recommend those materials back to students, depending on search queries. As successful as it is, this attempt fail short of creating a fully functional ontology that go beyond linking part of educational materials, but also represents the knowledge domain of a course. Another interesting approach was the one followed by Lau et al. [12]. They extract a domain ontology for a course based on the post on its online forum. While the techniques used worked, the quality of the domain ontology was heavily influenced by the (lack of) participation of the students in the forum. In these two examples, the links or

LAK19, March 4–8, 2019, Tempe, AZ, USA

ACM ISBN 978-1-4503-6256-6/19/03.
https://doi.org/10.1145/3303772.3303834

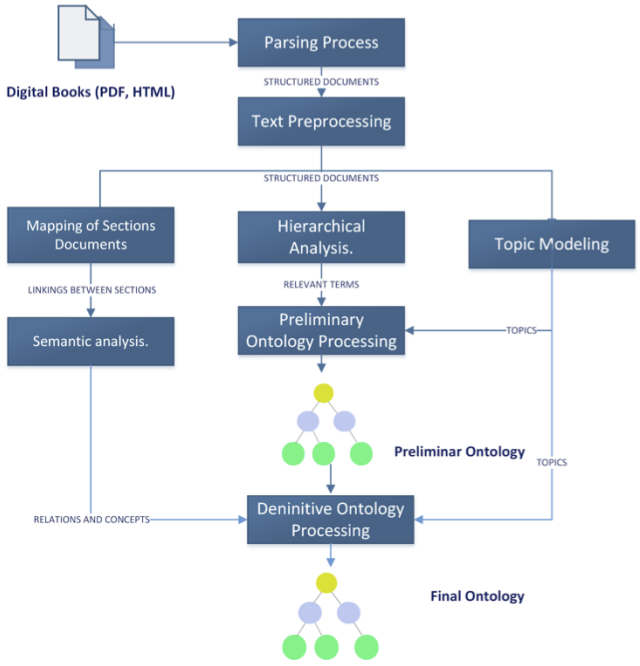

Figure 1: Automatic extraction of ontology from different sources process

ontologies automatically created were not designed to be changed or fixed by the domain expert, even in the case of computational error.

The main contribution of this work is the creation and evaluation of an automatic algorithm to extract course-centric ontologies from authoritative sources (digital textbooks, web tutorials or syllabus as digital text) that could be used by Learning Analytics tools and could be easily modified by the domain expert without the need to know about semantic technologies.

3 AUTOMATIC ONTOLOGY EXTRACTION

A process is proposed to generate ontologies that can represent the basis semantic relationships of a given domain, through topics detection from collections of semi-structured documents such as tutorials, digital books, and other educational resources. For clarity, this process is being divided in several steps (Figure 2) that are described in the following subsections.

3.1 Source Selection

The selection of relevant material is a manual step. The domain expert collects documents that she or he thinks that are relevant for the course. These documents could be digital textbooks, web tutorials, syllabus in digital format, among others. They could cover all or part of the topics of the course. Currently, the automatic extraction system is able to process Portable Document Format (PDF), Hyper Text Markup Language (HTML) or plain text files. If the original document is not in one of those formats (for example Microsoft Word), it needs to be converted. This is just a technical requirement that could be improved in the future.

Figure 2: Example of the PDF parser recognizing the structure of the document using the style attributes

3.2 Parsing

After the selection of documents for the course, different parsing algorithms (for each specific file format) are applied to extract the structure information and the raw text and to start building the abstract representations of each original documents. For HTML documents, a java process with Jsoup library is applied. This process has methods to analyze HTML content and obtain from it a tree structure, where each text section is a node. The methods expect that each chapter of a book is in a different file and that sections and subsections are labeled with specific CSS classes. For pdf documents, Java iText library is applied, to implement methods that can read the PDF sections. This library has methods that allow extracting the structure, take into consideration the style attributes (alignment, bold, italics, bullets, indentation and underlining), table of contents, and indexes present in the documents. Figure 2, present how the PDF parser can extract the structure of the document from the styles parameters of the text. As a final product, the parser divides the document into hierarchical parts (chapters, sections, subsections).

3.3 Text Pre-processing

For each section of the document, customary text mining techniques are applied for pre-processing. First words that are not useful (stopwords) are eliminated, then the text is tokenized an stemmed. Frequent words in textbooks and tutorials such as "exercises", "examples", "solutions", are also removed. Additionally, the domain expert can intervene to eliminate other words that could confuse the rest of process.

3.4 Topic Modeling

Once the text is clean, the Latent Dirichlet Allocation (LDA) statistical modelling tool [13] is applied to each section of the document in order to determine the most relevant topics for that section. LDA generates topics based on word frequency from a set of documents and it is particularly useful for finding reasonably accurate mixtures of topics within a given document set.

3.5 Hierarchical Analysis

Parallel to the topic modelling, the hierarchical structure extracted form the documents (chapters, sections and subsections) is used to establish a first relationship between domain concepts. These concepts are extracted from the title of the sections and subsections using keyphrase extraction [14] and Name Entity Recognizer (NER) [15] methods. A concept (class) is described by one or more relevant terms extracted in the titles of the chapters and subchapters. Each concept in the structure becomes a concept within the ontology, and the parent-child relationship (HasPart) between the elements becomes a relation in the ontology.

3.6 Mapping of Sections

Similarly to what Guerra at al. [11] did to link book sections, all parts of the documents are mapped to similar ones in other documents, using as a metric the similarity between the text of the two parts. For example, the content of the first section of the first document could be linked to the third section of the second document if the similarity between the text in both sections is higher than a threshold. To obtain the similarity metric value a semantic similarity algorithm is applied between all the parts of all documents. The result is a weighted list of links between different document parts and list of topics words associated with each document part. It expected that similar parts describe the same concept.

In this specific system, the cosine distance [16] was used to measure the semantic similarity between two document parts. The text of each part is represented as a multidimensional vector where the value for each dimension is the frequency of a given word in that text. The metric is the cosine of the angle between these two vectors. A thresholds for the minimum value of this metrics is set to establish similarity between each pair of document parts.

3.7 Preliminary Ontology Processing

The main task in this step is to create simple and compound concepts in a preliminary ontology to model the knowledge of the domain. This process take the concepts identified in the titles sections and sub-sections during the hierarchical analysis process and check if those concepts also appear in the list of topics associated with their corresponding text content. If a concept is identified in the title and the corresponding text, that concept is added to the ontology.

3.8 Semantic Analysis

To obtain more and deeper relationships between concepts than the ones present in the hierarchical analysis, a semantic analysis is run over the text content of each document part. There are several approaches to determine the conceptual relationships between words, mainly based on the assumption that concepts that are semantically related, tend to appear near one another text ([17]. For the semantic analysis and the relationships extraction, the system use Part-Of-Speech Tagging (POST) [18] and open information extraction (open IE) [19] techniques that conduct a linguistic analysis of sentences and paragraphs terms, verbs and proper names. First, each sentence is divided into a set of related clauses. Each clause is reduced to its minimum, producing a set of shorter sentence fragments. These fragments are then segmented into OpenIE triplet, which is grouped and prioritized according to the concepts and terms identified in the preliminary ontology. Figure 4 shows an example of a semantic relationships in a particular sentence.

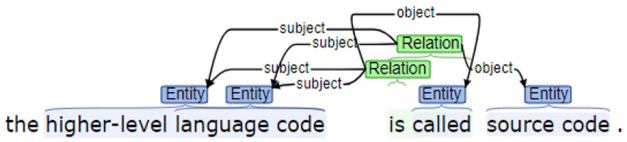

Figure 3: Semantic relationship in a sentence

3.9 Definitive Ontology Processing

The preliminary ontology results are associated with the relationships detected in the semantic analysis process. The preliminary ontology become a graph structure with hierarchical levels. This new ontology is represented with a graph in which nodes are relevant concepts that belongs to the domain of interest and edges are relations between the concepts.

This process also identifies possible problems in the ontology structure such as overgrowth depth of concepts in the graph and the existence of repeated terms. Finally, the definitive graph will be used to generate a domain ontology in standard Web Ontology Language (OWL) format and document parts are converted into HTML content. This is the normal final result of any ontology building process.

4 SYSTEM EVALUATION

To evaluate the technical aspects of the proposed automatic extraction process, as well to measure the quality of the resulting ontologies, the system was applied to two different courses: Programming Fundamentals and Digital Circuit Design.

4.1 Exemplary Ontology Extraction

First, for each course, a professor that teach the course was asked to select two textbooks that they use in the course or they know cover the content of the course. For Programming Fundamentals, the professor selected "Learn to Program with Python", and "Think Python". For Digital Circuit Design, the recommended books were "Fundamentals of Digital Logic with Verilog Design" and "Digital Design Principles and Practices". Second, the parsing algorithms were applied to the digital version of the books to obtain several text files corresponding to chapters, sections and subsections for each document. The subsections are considered as leaf nodes and contain raw text (the book content). Then to prepare the data and to create the documentary corpus, the text preprocessing routines were run over the individual document parts. Using Java Mallet library [1], multiples executions were run to obtain the LDA topic models. For set the parameter K, the number of topics to extract, the LDA algorithm, the midpoint between the average number of sections and the number of subsections of each book (n = 120 to 160) was used. The number of iterations of the LDA algorithm was set from 1200 to 1500.

[1]Java Mallet text processing library - http://mallet.cs.umass.edu/

The number of topics words to be extracted is a configurable parameter of the topic extraction algorithm. For this demonstrative runs, values between two and fifteen were used. The size of the final ontology varied according to this value. A smaller number of topic words, resulted in a smaller number of entities and relationships. To obtain a value between 60 and 70 concepts in the final ontology (as suggested usual for course-based ontologies) a value of 6 was finally selected.

The similarity between the different parts of the documents were calculated to generate links between those with the greatest similarity. As a result, a ranked list of links between documents parts was obtained, additionally to the topics associated with each link. Table 1 shows an example of the similarity detected between the sections of two Programming Fundamentals books.

Table 1: Similarity between sections of two Programming Fundamentals books.

Book1	Title	Book 2	Title	Sim.
bk1	Learning Python	bk2	Think Python	0.74
bk1-1-1	Software	bk2-18-11	Interfaces	0.26
bk1-1-3	Prog..with Python	bk2-2	The way of the program	0.49
bk1-2	Values and Variables	bk2-8-1	Multiple assignment	0.77
bk1-2-1	Integer Values	bk2-3-1	Values and types	0.94
bk1-2-2	Variables and Assignment	bk2-8-2	Updating variables	0.85
bk1-2-3	Identifiers	bk2-3-3	Variable names	0.92
bk1-2-4	Floating-point Types	bk2-6-1	Modulus operator	0.72
bk1-2-5	Control Codes	bk2-2-5	The first program	0.18
bk1-2-6	User Input	bk2-6-11	Keyboard input	0.84
bk1-2-7	The eval Function	bk2-6-11	Keyboard input	0.76
bk1-2-8	The print Function	bk2-7-9	Debugging	0.38

The hierarchical structure was constructed taking the concepts which belongs to titles and subtitles of the books sections that have the greatest similarity and shared the same topics. Only concepts that were present in both books were selected. During this process, only the topic words and entities detected in the titles of the chapters and sub-chapters were used, without considering the text information of all subsections. After this process, each term becomes a concept within the ontology, and the parent-child relation between terms becomes an relation in the ontology. A similarity cut-off values of 0.54, and 0,48 were selected for the Programming Fundamentals and Digital Circuit Design ontologies respectively.

For each of the document parts text, the Textrank library [2] was used to summarize the text, in order to find the most relevant sentences. An internal graph is constructed where the vertices represent each sentence in a document and the edges between sentences are based on content overlap, that is the number of words that 2 sentences have in common.

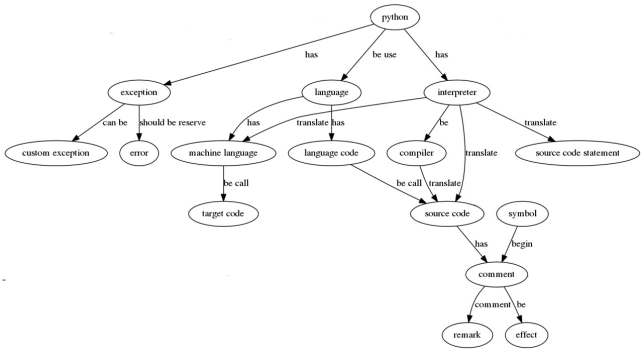

Figure 4: Programming Fundamentals preliminary ontology

Then, during the semantic analysis process, ClausIE [3] was applied to the summarized senteces to extract relations between terms.

Finally, the triples extraction process identified hundreds of triples from the main sentences from the documents, but only selected those that had terms present in the topic list associated to the each document parts. Then, to generate the definitive ontology, these terms were connected and added to the preliminary ontology. Figure 5 shows an extract from the final ontology for the Programming Fundamentals course. This ontology has 54 terms in a hierarchy in its first three levels and 61 terms in the first fourth levels. For the Digital Circuit Design course, the final ontology has 57 terms in the first three levels and 68 in the first four levels.

4.2 Evaluation

To evaluate the ontology quality, the same professors that recommended the books were asked to manually create an ontology. The creation process was guided by an ontology expert external to the process designing team. These manually generated ontologies were consider as the ground truth to evaluate the precision and recall of the automatically identified concepts. To compare the concepts between both ontologies, a process of stemming and similarity comparison was followed.

The precision was calculated through the as the percentage of concepts that were in the automatically generated ontology that also were present in the manually generated ontology compared with the total number of concepts in the automatically generated ontology. The recall was calculated as the percentage of concepts of the manually generated ontology that also were present in the automatically generated ontology.

The precision and recall for the Programming Fundamentals course were 72% and 59% respectively. The same quantities for the the Digital Circuit Design course were 67% and 57%, also respectively. These results suggest that the automatic extraction is able to produce a quite precise ontology that could be used for machine consumption at a fraction of the cost of the manual ontology generation.

[2]Textrank text summarization library - https://nlpforhackers.io/textrank-text-summarization/

[3]ClausIE Open Information Extractor - https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/software/clausie/

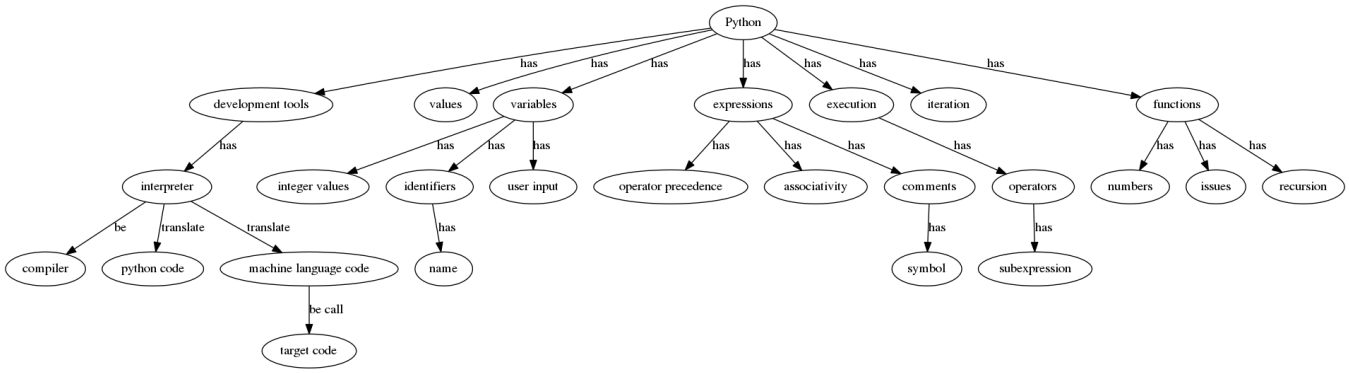

Figure 5: Programming Fundamentals automatically generated ontology (excerpt)

5 CONCLUSIONS AND FURTHER WORK

This work contributes a way to resolve the ontology-building cost barrier that limits the use of Intelligent Curriculum for Learning Analytics applications. With very little effort from the end-user, just recommending authoritative sources of learning materials such as digital books, an acceptable ontology of the domain can be automatically created. In the evaluation of our methodology, two books were enough to create an ontologies that in average were 70% precise and capture more than 50% of the concepts generated manually by an expert. The generated ontologies could be used by other automatic systems to recommend materials, provide feedback to students or to monitor discussion forums, for example.

While this work offers a working system to automatically generate course-based ontologies, it also presents several limitations that could be overcome with further work. The end-user could play with the different model parameters, specially for the LDA model, to obtain more inclusive ontologies, sacrificing some precision, or vice versa. Currently these parameters are fixed in at design time. Another limitation of this work is the lack of consideration of the links into the ontology quality. A graph-based similarity measure could be used to better understand the relationships between the automatic and manual generated ontologies.

The final goal of this work, to be realized in further work, is the creation of an application to support the semi-automatic generation of Intelligent Curriculum. In this envision application, the instructor will add learning content, the system will recommend a preliminary ontology and the instructor will be able to improve it via additional material or direct manipulation. Such a tool will be able to open many opportunities for the development of more sophisticated and useful Learning Analytics solutions.

REFERENCES

[1] George Siemens. "Learning Analytics: Envisioning a Research Discipline and a Domain of Practice". In: *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. LAK '12. Vancouver, British Columbia, Canada: ACM, 2012, pp. 4–8.

[2] George Siemens and Ryan S. J. d. Baker. "Learning Analytics and Educational Data Mining: Towards Communication and Collaboration". In: *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*. LAK '12. Vancouver, British Columbia, Canada: ACM, 2012, pp. 252–254.

[3] Wenhuan Lu and Jianguo Wei. "Dynamic visualization of Evolutionary Curricula Model". In: *2010 International Conference on Educational and Information Technology*. Vol. 2. 2010, pp. V2-484-V2-488.

[4] Dragan Gaaevic et al. *Model Driven Architecture and Ontology Development*. Berlin, Heidelberg: Springer-Verlag, 2006.

[5] Yu-Liang Chi. "Ontology-based Curriculum Content Sequencing System with Semantic Rules". In: *Expert Syst. Appl.* 36.4 (May 2009), pp. 7838–7847.

[6] Aysu Ezen-Can et al. "Unsupervised Modeling for Understanding MOOC Discussion Forums: A Learning Analytics Approach". In: *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*. LAK '15. Poughkeepsie, New York: ACM, 2015, pp. 146–150.

[7] Erik Duval. "Attention Please!: Learning Analytics for Visualization and Recommendation". In: *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. LAK '11. Banff, Alberta, Canada: ACM, 2011, pp. 9–17.

[8] Sabrina Ziebarth, Nils Malzahn, and H. Ulrich Hoppe. "Using Data Mining Techniques to Support the Creation of Competence Ontologies". In: *Proceedings of the 2009 Conference on Artificial Intelligence in Education*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2009, pp. 223–230.

[9] Martin Hepp. "Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies". In: *IEEE Internet Computing* 11.1 (Jan. 2007), pp. 90–96.

[10] Wilson Wong, Wei Liu, and Mohammed Bennamoun. "Ontology Learning from Text: A Look Back and into the Future". In: *ACM Comput. Surv.* 44.4 (Sept. 2012), 20:1–20:36.

[11] Julio Guerra, Sergey Sosnovsky, and Peter Brusilovsky. "When One Textbook Is Not Enough: Linking Multiple Textbooks Using Probabilistic Topic Models". In: *Scaling up Learning for Sustained Impact*. Ed. by Davinia Hernández-Leo et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 125–138.

[12] Raymond Y. K. Lau et al. "Toward a Fuzzy Domain Ontology Extraction Method for Adaptive e-Learning". In: *IEEE Trans. on Knowl. and Data Eng.* 21.6 (June 2009), pp. 800–813.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 993–1022.

[14] Kazi Saidul Hasan and Vincent Ng. "Automatic Keyphrase Extraction: A Survey of the State of the Art". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: ACL, June 2014, pp. 1262–1273.

[15] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, 2005, pp. 363–370.

[16] Anna Huang. *Similarity Measures for Text Document Clustering*. 2008.

[17] Alexander Maedche and Steffen Staab. "Mining Ontologies from Text". In: *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*. EKAW '00. London, UK, UK: Springer-Verlag, 2000, pp. 189–202.

[18] Kristina Toutanova et al. "Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. NAACL '03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 173–180.

[19] Anthony Fader, Stephen Soderland, and Oren Etzioni. "Identifying Relations for Open Information Extraction". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011, pp. 1535–1545.