



AUTOMATIC TEXT SUMMARISATION OF CASE LAW USING GATE WITH ANNIE AND SUMMA PLUG-INS

C. T. Aghaunor¹ and G. O. Ekuobase^{2,*}

^{1, 2} DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF BENIN, BENIN CITY, EDO STATE, NIGERIA

E-mail addresses: ¹ tabitha.ghaunor@gmail.com, ² godspower.ekuobase@uniben.edu

ABSTRACT

Legal reasoning and judicial verdicts in many legal systems are highly dependent on case law. The ever increasing number of case law make the task of comprehending case law in a legal case cumbersome for legal practitioners; and this invariably stifles their efficiency. Legal reasoning and judicial verdicts will therefore be easier and faster, if case law were in abridged form that preserves their original meaning. This paper used the General Information Extraction System Architecture approach and integrated Natural Language Processing, Annotation, and Information Extraction tools to develop a software system that does automatic extractive text summarisation of Nigeria Supreme Court case law. The summarised case law which were about 20% of their original, were evaluated for semantic preservation and has shown to be 83% reliable.

Keywords: Case law, text summarisation, text engineering, text annotation, text extraction.

1. INTRODUCTION

The computer has remained notable for speed, information processing, exchange and storage but still unable to comprehend and interpret the information it is made to store, manipulate or exchange. With the high information overload across several domains, the task of processing and extracting meaningful facts from “these sea” of information is increasingly laborious, inefficient and ineffective [1–3]. Individuals and organizations are finding an increasing gap between the acquisition of information and their meaningful use, despite the increasing influx and access to the information [1] due to the inability of the computer – the core information processing tool – to comprehend and interpret the information. This may account for the poor decision bedeviling every aspect of the world in recent times; as humans have to study, understand and extract useful facts for decision making from the sea of information; a task that would have been more efficient and reliable if computers could comprehend the information and work in cooperation with humans in extracting and interpreting required facts from available information [4–6].

As regards decision making, poor decision in law will not only be a disaster to the legal profession but also to the society it controls. The legal profession, world over, keeps track of their legal information in form of statutes, legislation and case law [7]. Of these legal recordings, the most active is case law as legal decisions are inherently case based – “stare decisis” [8]. For efficient and quality legal decision therefore, the computer must be made to comprehend case law and assist legal practitioners in the task of extracting relevant facts from available information. Making the computer comprehend and summarise information (i.e. automatic summarisation), is the essence of semantic annotation and extraction.

A summary is a text produced from one or more texts, which conveys important information in the original text, and is of a shorter form. Radev, *et al* [9] defined summary as “a text that is produced from one or more texts, that conveys important information in the original text(s), and that is not longer than half of the original text(s) and usually, significantly less than that”. Automatic text summarization is the task of using computers to produce a concise and fluent summary while preserving key information content and overall meaning [10]. The goal of automatic text

* Corresponding author, tel: +234 – 806 – 495 – 1845

summarisation is to present the source text as a shorter version with the semantics retained [11, 12]; that is to express the contents of a document in a concise and meaningful form that meet user needs. The most important advantage of using a summary instead of the original document is that it saves reading and comprehension time.

Tools that provide timely access to and digest of varying information sources are necessary in order to alleviate the effect of information overload. These concerns have sparked interest in the development of automatic summarisation systems or summarisers. Such systems are designed to take one or more documents or information piece and produce a concise and fluent summary of the most important information [11]. Summarisers achieve this by identifying the most important sentences in the input document(s), and stringing them together to form a summary.

Text Summarisation methods can be classified into extractive and abstractive summarisation [10, 11, 13]. Other classifications include indicative, informative and critical summarizations [12, 14]. This paper is centred on extractive summarisation – verbatim extraction or copying of parts of document(s) that retains the semantics of the document(s) in context. This is most suited to case law where the original words and intent of judges must be preserved.

A good way for computers to perform this task is to have documents in a state where their meanings are explicitly understood and can be interpreted by software processes rather than being implicitly interpretable by humans only [15]. Making the computer understand and interpret case law is particularly crucial as the legal domain is usually case based. Though this will reduce the stress of advocacy, expedite legal decision processes and bring about improved efficiency and accuracy of adjudication; it is nontrivial [16]. This work dedicated itself to this nontrivial task of annotation, extraction and summarisation of case law in the context of case element. Case elements are the critical components of case law that are collectively sufficient to describe case law; and are very useful to legal practitioners for legal reasoning and judicial verdicts.

2. RELATED WORKS

Popov, *et al* [17] described an approach towards semantic web information extraction and presented a model for semantic content enrichment. The model was implemented on a system called the Knowledge and Information Management (KIM) platform. KIM performs information extraction based on ontology

and a massive knowledge base. The Information Extraction (IE) process in KIM was based on the General Architecture for Text Engineering (GATE) platform and it directly reused some of GATE's generic Natural Language Processing (NLP) components. The system's information extraction approach was based on the recognition of Named Entities (NEs) with respect to formal upper-level ontology. However, Popov, *et al* [17] were only concerned with named entity extraction and not text summarisation.

The work of Schilder and McCulloh [18] was centred on temporal information extraction from legal documents. The work analysed the kinds of temporal information that could be found in the diverse types of legal documents; by providing a comparison of the different legal document types (case law, statute or transactional documents). Although, the work focused on temporal information in legal text, how the information could be automatically extracted and how one could do reasoning with the extracted temporal information in order to add more value to the document; the work carried out extraction without annotation and thus not amenable to machine comprehension.

Wiebe, *et al* [19] described the manual annotation of corpus based on opinions, emotions, and sentiments amongst other private states in language. The research [19] stemmed from the desire to aid analysts in government and political domains to automatically track attitudes and feelings of people about happening events from news and online forums. The work presented multiple answers to non-factual multi-perspective questions based on opinions from different sources. Annotation gold standard was realized manually and they made use of GATE which used the gold standard as basis to annotate other document sentences. However, the work's IE was abstractive and not extractive.

The work of Soria, *et al* [20] focused on the automatic extraction of Italian law documents. The work made use of semantic computational tool called Semantic Annotation for LEgal Management (SALEM) and NLP techniques to perform the task of classifying Law paragraphs of Italian documents according to their regulatory content and then extracted the relevant texts fragments corresponding to specific semantic roles that are relevant for each law document. Soria, *et al* [20] however, did not semantically enrich or annotate Italian legal documents and hence scalable automatic information extraction from such documents is stifled. Also, the work did not concern itself with individual summarization of case law.

Park, *et al* [21] described a semi-automatic annotation tool for building large dependency tree-tagged corpus. The research was carried out in order to develop robust statistical NLP systems by annotating corpora with lots of linguistic information. Specifically, they designed and implemented an annotation tool called "PPeditor" which was used to establish a Korean dependency tree-tagged corpus. Park, *et al* [21] also annotated text with segmentation of word phrases (called *eojeols* in Korean language), Parts of Speech (POS) tags and chunk annotation. However, the work restricted itself to annotation of Korean text for error detection and correction purposes only.

Wyner [16] discussed an experiment that used NLP tools on case law to produce annotated texts in order to support IE. The author made use of the GATE framework and applied its NLP components but made use of limited number of case elements. Besides, Wyner [16] made no attempt to either do extractive summarization or output the summary; the work basically ended with annotation.

Sapkota, *et al* [22] described a framework for semantic annotation and automatic extraction of regulatory text. The work evolved from the need to convert regulatory texts to machine interpretable models in order to enhance the automatic speedup of the Compliance Management (CM) process [22]. To handle the problem encountered by different document layout, they proposed a semi-automatic specification of the relevant document format used by the regulatory bodies with which they developed a gold standard. Although the work made use of the GATE platform and applied its NLP components, it was not directed at case law summarisation.

Albukhitan and Helmy [23] presented automatic annotation of Arabic Web resources relating to food, nutrition and health domain. The research was aimed at producing high quality web content in the stated domains through semantic annotation of their web sources. The work made use of related Arabic OWL ontology and linguistic patterns to discover and extract relevant relationships among NEs in Arabic web resources. Overall, the work leveraged semantic web technologies to serve the Arabic language and produce semantically annotated web documents for the targeted domains in an automatic manner. This work however, restricted itself to IE of non-legal Arabic documents.

Barkschat [24] provided an ontology based knowledge model for semantic data extraction on domain specific data. Barkschat [24] used formal knowledge representation to mimic manual extraction and

developed a system for improved data extraction. The work which was based on the Extract-Transform-Load process realised an Ontology Based Information Extraction (OBIE) system. OBIE supported automatic relation extraction; but its output lacks typographic structures (i.e. it is still an unstructured text) and is not amenable to machine comprehension.

Ekuobase and Ebietomere [25] built ontology for Nigerian case law with the intent of expediting the Nigerian Judicial process but had their attention directed at Information Retrieval (IR) of case law. Our focus is IE, particularly, extractive summarisation of case law. Thus, this paper furthers the intent of Ekuobase and Ebietomere [25] by targeting the abridged equivalent of Nigerian case law. It is obvious therefore that scarcely has any previous research work dedicated itself practically to the extractive summarisation of case law – a gap this paper attempts to fill.

3. MATERIALS AND METHOD

This paper adopted the General IE System Architecture approach – the defacto approach to text annotation and extraction [26]. The annotation and summarization of case law were based on case elements. The case elements considered include: case title, name of court, date of judgment, judge(s), suit number, parties in court, lead judge – where there exists more than one case decider, fact of the case, cause of action etc. In particular, annotation of the selected Nigeria Supreme Court case law was performed using GATE with A Nearly New Information Extraction System (ANNIE) components while extractive summarisation of the annotated case law was carried out using GATE with SUMMA plug-ins [27, 28]. The study made use of 72 Nigeria Supreme Court electronic case law.

3.1 Annotation of Case Elements in Case Law

To annotate case elements of the selected case law, the case law corpus was created and loaded on the GATE platform. To create case law corpus, Language Resources was right clicked on the Resource Tree in the GATE Developer Main View which displays on launching GATE; as shown in Figure 1. Then, New+GATE corpus was selected and thereafter, parameters for the corpus in the Parameter Dialog Box were set. The "ok" button was then clicked. On successful creation of the corpus, the corpus name displayed on the Language Resources menu of the Resource Tree in the main view; as shown in Figure 2.

To load the case law into the created corpus, the created corpus (Case Law Corpus) on the Language Resources menu of the Resource Tree, was right clicked.

"Populate" button was then selected and the directory where the case law were stored on the Dialog Box that appeared was supplied. The "ok" button was clicked to complete the task. On successfully loading the case law in the Language Resource menu of the Resource Tree; as shown in Figure 3.

Subsequently, the required annotation "Processing Resources" beneath the created case law corpus in the Resource Tree were loaded; by repeatedly right clicking and selecting New+Additional Required Resource until all required processing resources were loaded. The loaded processing resources immediately displayed on the Processing Resources menu in the Resource Tree; as shown in Figure 3. The loaded processing resources were: ANNIE Sentence Splitter (for sentence segmentation), ANNIE English Tokenizer (for tokenization), ANNIE POS Tagger (for POS tagging) and ANNIE Gazetteer (for entity and relation detections).

The required SUMMA Processing Resources were also loaded following the same iterative process as that of the ANNIE Processing Resources. When successfully loaded, the SUMMA Processing Resources were displayed in the GATE Developer main view as shown in Figure 4. The selected SUMMA Processing Resources were: SUMMA NEs Statistics, SUMMA Position Scorer, SUMMA Sentence Document Similarity, SUMMA Normalize Vector, SUMMA Term Frequency Filtering, SUMMA Vector Computation, SUMMA First Sentence Similarity, SUMMA Sentence Term Frequency Scorer, SUMMA Simple Summarizer and SUMMA Extract Exporter. The tasks of the selected SUMMA Processing Resources are detailed in [27].

The next task was to "Run" the resources on the loaded cases. This was done by right clicking "Applications" button on the Resource Tree of the GATE Developer Main View. Then, Create New Application+Corpus Pipeline was selected. Thereafter a dialog box with the parameters for the new corpus pipeline was displayed and the "ok" button on it was clicked to complete the task. This immediately created a corpus pipeline below the Application message in the Resource tree as shown in Figure 5. The created Corpus Pipeline was then populated with the loaded Processing Resources and

the Corpus Pipeline's parameters were set as required to achieve the desired annotation and summarisation. To populate the Corpus Pipeline, the Corpus Pipeline was right clicked and the "shows" button that displayed thereafter was selected. The loaded Processing Resources were then displayed for selection in the order they will be Run; as shown in Figure 5.

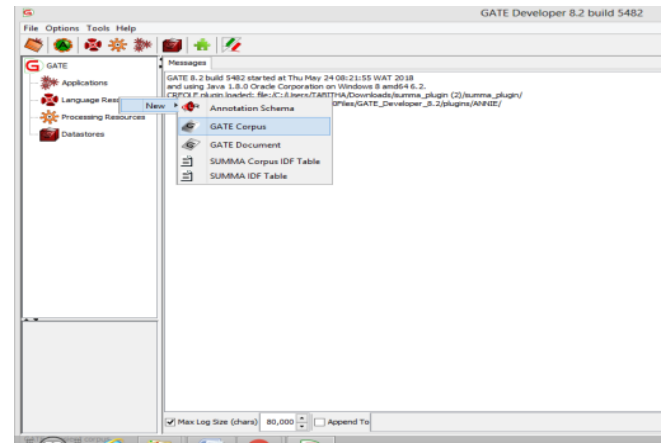


Figure 1: GATE Developer Main View

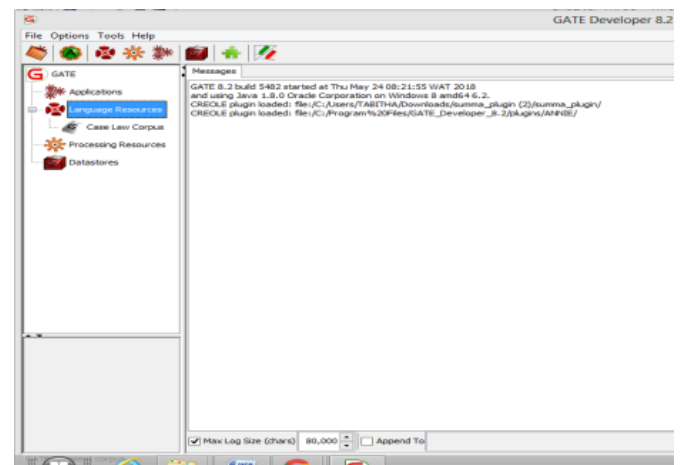


Figure 2: GATE Developer Main View Showing the Created Case Law Corpus

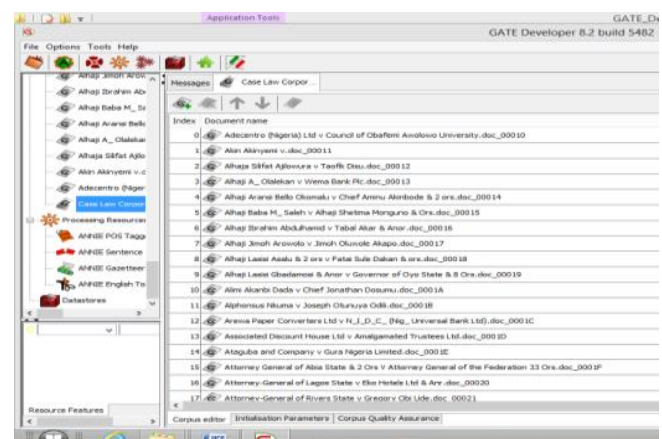


Figure 3: GATE Developer Main View Showing the Loaded Case Law and ANNIE Resources

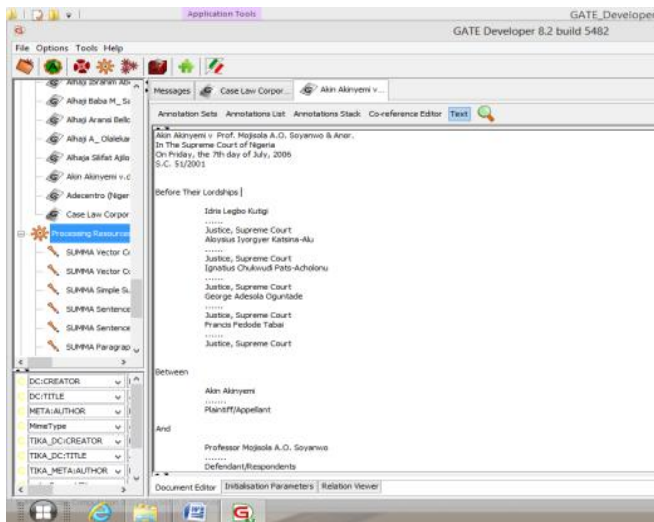


Figure 4: GATE Developer Main View Showing the Selected SUMMA Processing Resources

The needed summarisation annotation resources were then moved in the order of execution from the left pane to the right pane (as shown in Figure 5) and thereafter, the Run button at the center of the dialog box in Figure 5, was clicked. When successful, a new annotated version of the loaded case law automatically displayed on top of the original case law on the Language Resources menu of the Resource Tree.

3.2 Extractive Summarisation of the Annotated Case Law

The summarisation process involved creating a new corpus pipeline on the Application menu of the Resource Tree. To achieve this, "Applications" in the Resource Tree was clicked. Create New Application+Corpus Pipeline was then selected and thereafter, a dialog box with parameters for the new corpus pipeline displayed and the "ok" button on it was clicked to complete the task. A new corpus pipeline was then created below the Application menu as shown in Figure 6. This created Corpus Pipeline was thereafter populated by loading the SUMMA Simple Summariser and setting its parameters.

Next, the "Run this Application" button at the bottom of the dialog box was clicked to complete the task of populating the Corpus Pipeline. The summarised versions retain the old name of the cases with the string "XXXXX Name" appended as shown in Figure 6. Where "XXXXX" is an arbitrary five digit hexadecimal numeral and Name is any variable name given to the annotation set in the SUMMA Simple Summariser by the Application Developer.

Finally, the summarised version of each case law in the corpus was exported to disk where they can be displayed using any browser or Notepad. The summarised versions can be further processed by the machine depending on application need since the summarised versions are fully structured.

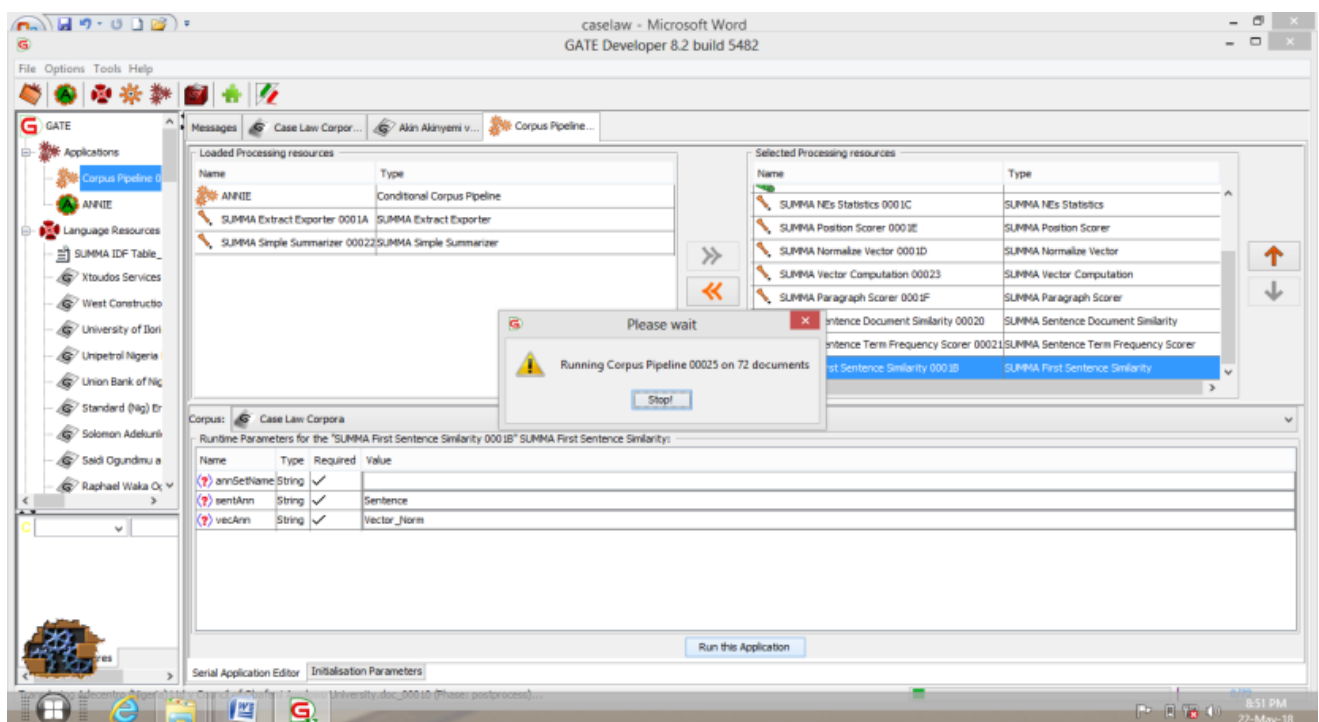


Figure 5: GATE Developer Main View Showing Loaded Resources Ready for Selection and Running

The export process involved creating a new corpus pipeline on the Application menu of the Resource Tree. To achieve the export process, "Application" was clicked on the Resource Tree in the GATE Developer Main View. Create New Application+Corpus Pipeline was then selected, thereafter, a dialog box with parameters for the new corpus pipeline displayed and the "ok" button on it was clicked to complete this task. A new corpus pipeline was then created on the Application menu of the Resource Tree as shown in Figure 7. This created Corpus Pipeline was also populated by loading the SUMMA Extract Exporter and

setting its parameters based on the output details of the SUMMA Simple Summariser. The "Run this Application" button at the bottom of the dialog box was then clicked to export the summarised case law corpus to the directory of choice on disk where they can be stored.

Figure 8 shows the summarised version of the case law corpus ready for opening in Avast Safe Zone Browser. It should be noted that each of the summarised versions of the case law was reduced by about 80%.

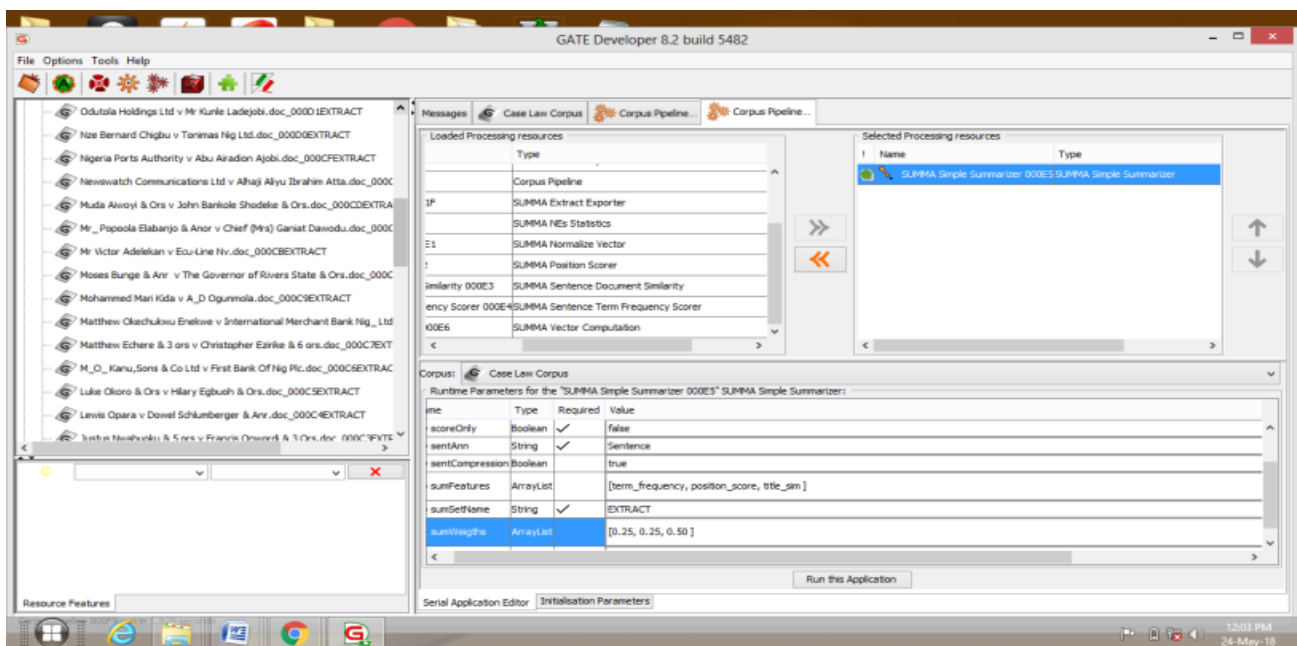


Figure 6: GATE Developer Main View Showing Summarised Case Law

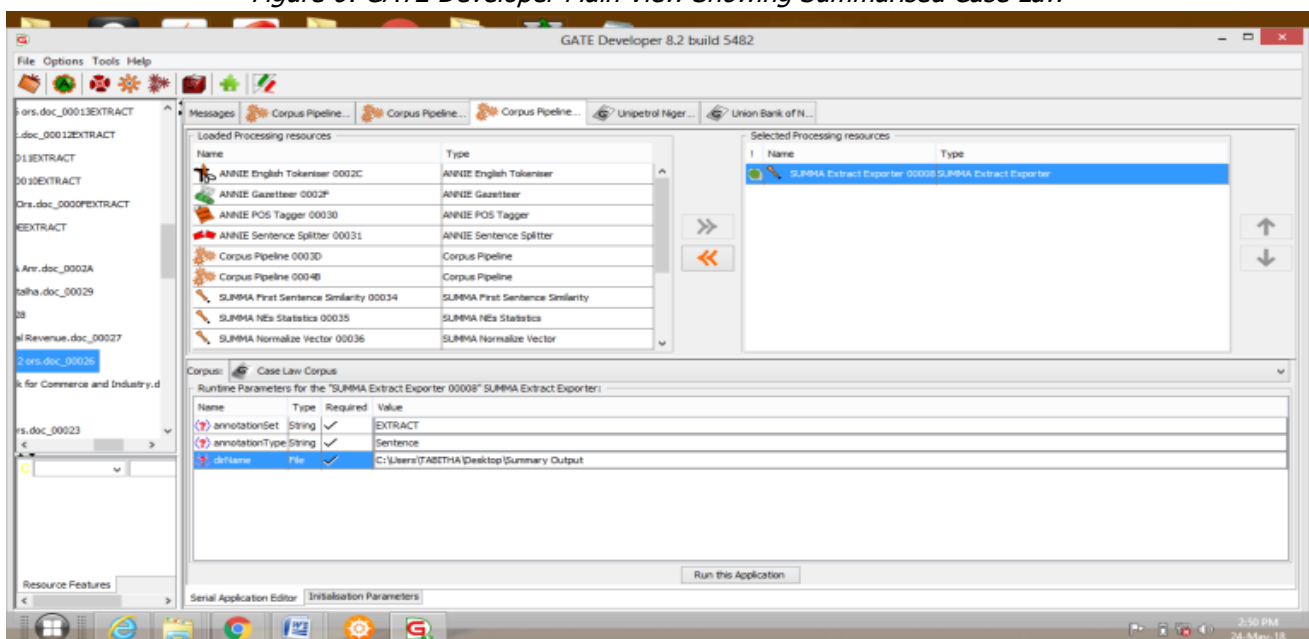


Figure 7: GATE Developer Main View Showing the Process of Exporting Summarised Case Law to Disk

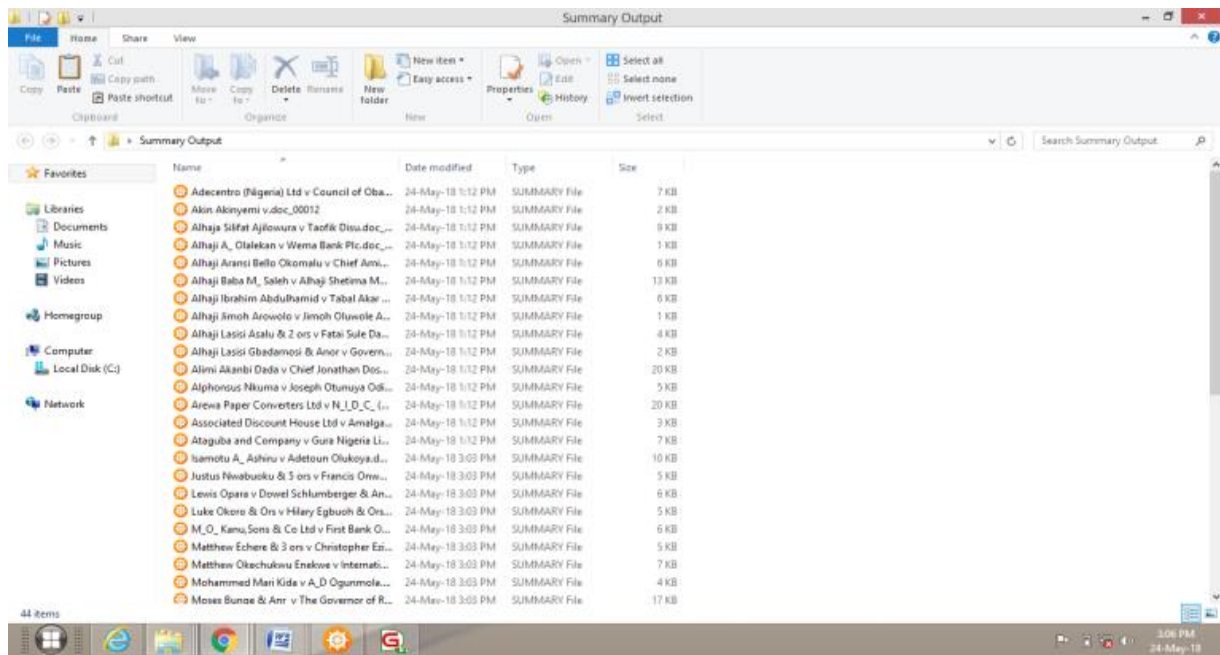


Figure 8: Output of Summarised Nigeria Supreme Court Case Law Corpus

4. EVALUATION OF THE CASE LAW'S ANNOTATION FOR SUMMARISATION

A very critical part of IE is the evaluation of the annotated text on which extraction was done. The importance of evaluation in text engineering stems from the fact that what cannot be measured and expressed in either quality or quantity is inconsequential to man and oftentimes cannot be relied upon. Commonly, processes and operations in IE and IR are measured for purposes of dependency and trust using metrics such as Precision, Recall and F-measure [28, 29]. Consequently, this research paper measured the Precision, Recall and F-measure of the case law's annotation for extractive summarisation using GATE; since GATE is the platform of annotation, extraction and summarisation.

4.1 Annotation Evaluation in GATE

GATE is a complete text engineering tool not only because it supports most text engineering processes but also because it enables the processes, artifacts and systems built on it, to be evaluated for performance quality [28]. A veritable tool in GATE for evaluating annotation including those for IE is the Annotation Diff Tool (ADT). ADT is able to calculate the Precision, Recall and F-measure of the annotated text under evaluation according to three different criteria of strict, lenient and average [28]. The Strict measure considers all partially correct responses as incorrect (spurious), the Lenient measure considers all partially correct responses as correct

while the Average measure allocates half weight to partially correct responses (i.e. it takes the average of strict and lenient). These metrics for evaluating IE systems are defined as follows [28]:

(i) Strict Criteria

$$Precision = \frac{Correct}{Correct + Spurious + Partial} \tag{1}$$

$$Recall = \frac{Correct}{Correct + Missing + Partial} \tag{2}$$

$$F - Measure = \frac{(\beta^2 + 1)Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{3}$$

(ii) Lenient Criteria

$$Precision = \frac{Correct + Partial}{Correct + Spurious + Partial} \tag{4}$$

$$Recall = \frac{Correct + Partial}{Correct + Missing + Partial} \tag{5}$$

$$F - Measure = \frac{(\beta^2 + 1)Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{6}$$

(iii) Average Criteria

$$Precision = \frac{Correct + \frac{1}{2}Partial}{Correct + Spurious + Partial} \tag{7}$$

$$Recall = \frac{Correct + \frac{1}{2}Partial}{Correct + Missing + Partial} \tag{8}$$

$$F - Measure = \frac{(\beta^2 + 1)Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{9}$$

In all, β reflects the weighting of precision vs. recall. However in GATE's ADT, as is often the case, β was set to 1.

4.2 Evaluating Case Law’s Annotation for Summarisation using GATE’s ADT

This process of evaluation follows the process of annotation as described in Section 3.1. Thereafter, the ADT smart icon (a green-white-red-like flag) appearing at the top of the GATE’s Developer Main View as shown in Figure 9 was clicked. An ADT Dialog Box containing the annotated case law then displayed. Thereafter, Algorithm1 was performed on the annotated case law.

Algorithm 1: Algorithm for Evaluating Case Law’s Summarisation Annotation using GATE’s ADT

Step 1: Select next annotated case law

Step 2: Select the annotation set of the case law in Step 1

Step 3: Select “Sentence” – the annotation type

Step 4: Select “Compare” and wait for performance result to appear

Step 5: If more case law go to *Step 1* else go to *Step 6*

Step 6: Stop

The performance result for an annotated case law after evaluation is shown in Figure 9. The result in Figure 9a is about the best annotated case law result while the result in Figure 9b is about the worst annotated case law result. For conciseness, only these two results were shown since they defined the performance range of the summarisation annotation necessary and sufficient for result interpretation and discussion.

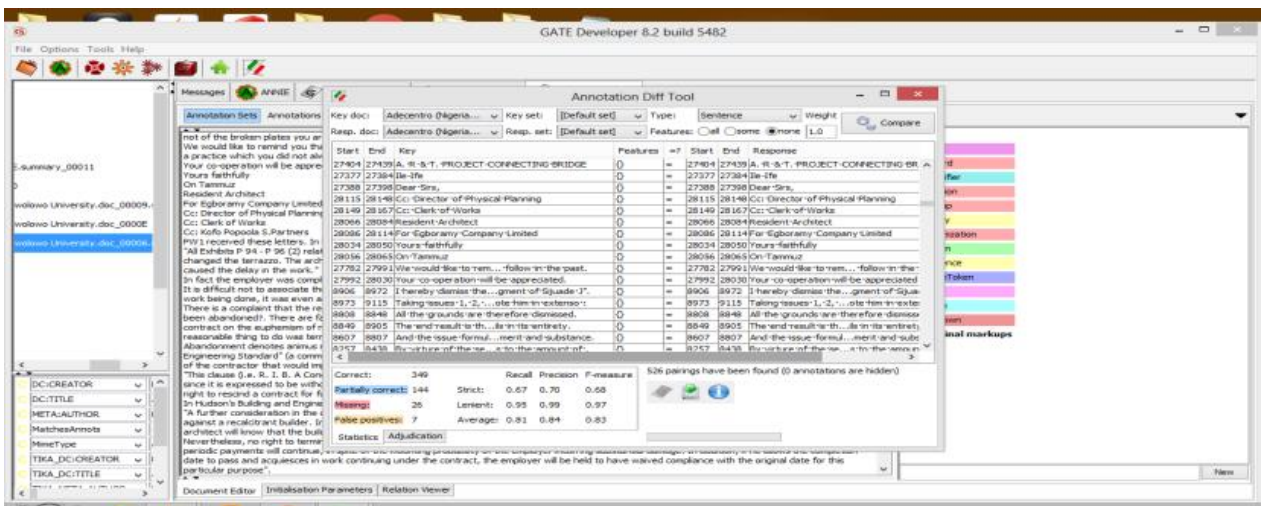


Figure 9: GATE Developer Main View Showing a Typical ADT Process

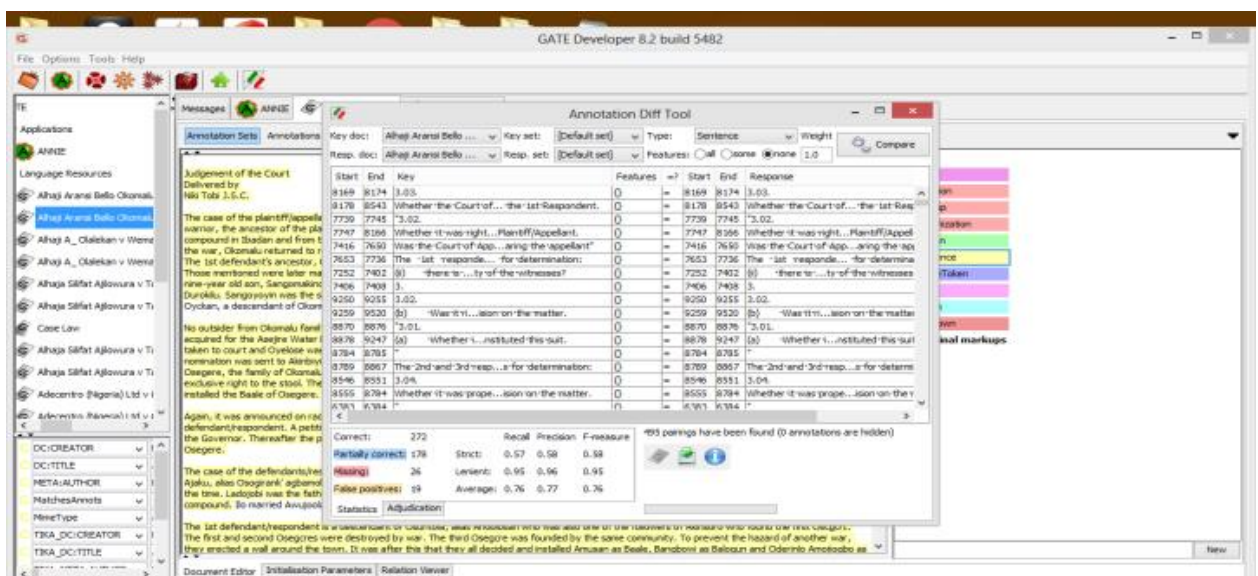


Figure 9b: GATE Developer Main View Showing a Typical ADT Process

Table 1: Performance Range of Automatic Summarisation Annotation of Nigerian Case Law

<i>BEST</i>	Recall	Precision	F-Measure	<i>WORST</i>	Recall	Precision	F-Measure
Strict	0.67	0.70	0.68		0.57	0.58	0.58
Lenient	0.95	0.99	0.97		0.95	0.96	0.95
Average	0.81	0.84	0.83		0.76	0.77	0.76

4.3 Result Interpretation and Discussion

Table 1 captures about the best and about the worst performance of the automatic summarisation annotation of the selected Nigerian case law using GATE with ANNIE and SUMMA plug-ins.

A strict recall of 0.67 means that 67% of the sentences, words and phrases in the case law were correctly and completely annotated as it should be annotated; a lenient recall of 0.95 means 95% of the sentences, words and phrases in the case law were annotated while an average recall of 0.81 means roughly 81% of the sentences, words and phrases in the case law were correctly annotated as it can be annotated.

A strict precision of 0.58 means that 58% of the annotated sentences, words and phrases in the case law were correctly and completely annotated as it should be annotated; a lenient precision of 0.96 means 96% of the annotated sentences, words and phrases in the case law were annotated correctly while an average precision of 0.77 means roughly 77% of the annotated sentences, words and phrases in the case law were correctly annotated as it can be annotated.

A strict F-measure of 0.68 means that 68% of the annotated sentences, words and phrases in the case law were of excellent annotation quality; a lenient F-measure of 0.97 means that 97% of the annotated sentences, words and phrases in the case law were of fair annotation quality while an average F-measure of 0.83 means that 83% of the annotated sentences, words and phrases in the case law were of good annotation quality. The average F-measure best captures the system's performance as it mitigates outliers.

The implication of this result is that the developed summariser is capable of 83%, but guarantees 76%, retention of the original case law's meaning. The summarised case law is not 100% in meaning compared to their original version as expected. The reason for this is probably due to the poor support of the tools used with respect to the indigenous names and culture of the Nigerian people as it affects her legal system.

5. CONCLUSION

Legal reasoning and judicial verdicts is highly dependent on case law. This ever increasing number of case law make the task of comprehending case law cumbersome even for experienced legal practitioners and this stifles their efficiency. This paper adopted the General IE Systems Architecture approach and deployed GATE platform with ANNIE and SUMMA plug-ins for automatic extractive text summarisation of some Nigeria Supreme Court case law. The automatic summarised case law which abridged the original case law by about 80% was established to be 83% reliable in the semantic preservation of its original version in the context of case elements. The result calls for creation of indigenous plug-ins to existing text engineering tools.

5.1 Suggestion for Further Studies

Efforts should be directed towards the development of automatic NLP, annotation and IE tools indigenous to a people's culture particularly as it affects their legal system. This could be in the form of building, or the formulation of knowledge to help build, application plug-ins for text engineering platforms like the GATE platform.

6. REFERENCES

- [1] Sanchez, M.F. "Semantically Enhanced Information Retrieval: An Ontology-based Approach", PhD Thesis, Universidad Autonoma De Madrid, 2009.
- [2] Sudeepthi, G., Anuradha, G. and Babu, M.S. "A Survey on Semantic Web Search Engine", *International Journal of Computer Science Issues IJSCI*, Vol. 9, Number 1, 2012, pp. 241-245.
- [3] The Economist Intelligence Unit. "The Evolving Role of Data in Decision-making", *EIU_CSC EMC_Data WEB*. In dd, 2013, pp.1-3.
- [4] Berners-Lee, T., Hendler, J. and Lassila, O. "The Semantic Web", *Scientific American*, Vol. 5, Number 284, 2001, pp. 34-43.
- [5] Guha, R., McCool, R. and Miller, E. "Semantic Search", *Proceedings of the Twelfth International Conference on the World Wide Web*. Hungary, 2003, pp. 700-709.
- [6] Wei, W., Barnaghi, P.M. and Bargeila, A. "Search with Meanings: An Overview of Semantic Search

- Systems*", School of Computer Science, University of Nottingham Malaysia Campus, 2008.
- [7] Lokulo-Sodipe, J. Akintola, O. and Adebamowo, C. "Legal Basis for Research Ethics Governance in Nigeria", *Introduction to the Legal System of Nigeria*, 2014, Vol. 2, pp. 73-77.
- [8] Aniekwu, N.I. "Legal Methodology and Research in Nigeria: An introduction", Mindex Publishing, 2010.
- [9] Radev, D.R., Hovy, E. and McKeown, K. "Introduction to the Special Issue on Summarization", *Computational Linguistics*, Vol. 28, Number 4, 2002, pp. 399-408.
- [10] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B. and Kochut, K. "Text Summarization Techniques: A Brief Survey", *In Proceedings of arXiv*, USA: ACM, 2017, pp. 3-9.
- [11] Nenkova, A. and McKeown, K. "Automatic Summarization", *International Conference on Language Resources and Evaluation (LREC 06)*, 2011, pp. 2-7.
- [12] Samrat, B. "Text summarization: An Overview", *Institute of Technology Journal*, Vol. 3, Number 2, 2014, pp. 2-6.
- [13] Gupta, V. and Lehal, G.S. "A Survey of Text Summarization Extractive Techniques", *Journal of Emerging Technologies in Web Intelligence*, Vol. 2, Number 3, 2010, pp. 10-11
- [14] Lal P. "Text Summarization", PhD Thesis, Imperial College London, 2002.
- [15] Bechhofer, S., Carr, L., Goble, C., Kampa, S. and Miles, T. "The Semantics of Semantic Annotation", *In Proceedings: On the Move to Meaningful Internet Systems, Confederated International Conferences*, Springer-Verlag, 2002, pp. 1152-1166.
- [16] Wyner, A. "Towards Annotating and Extracting Textual Legal Case Elements", *Proceedings of the Workshop LOAIT*, Vol. 19, Number 1-2, 2010, pp. 173-183.
- [17] Popov, B., Kiryakov, A., Ognyanoff, D., Manov D. and Kirilov, A. "KIM: a Semantic Platform for Information Extraction and Retrieval", *Natural Language Engineering*, Vol. 10, 2004, pp. 375-392.
- [18] Schilder, F. and McCulloh, A. "Temporal Information Extraction from Legal Documents", *In Proceedings of Dagstuhl Seminar on Annotating, Extracting and Reasoning about Time and Events*, 2005, pp. 5-9.
- [19] Wiebe, J., Wilson, T. and Cardie, C. "Annotating Expressions of Opinions and Emotions in Language", *Language Resources and Evaluation*, Vol. 1, Number 2, 2005, pp. 165-210
- [20] Soria, C., Bartolini, R., Lenci, A., Montemagni, S. and Pirrelli, V. "Automatic Extraction of Semantics in Law Documents", *In Proceedings of the V Legislative XML workshop*, 2007, pp. 253-266.
- [21] Park, E., Kim, J., Kim, C. and Kim, Y. "Semi-Automatic Annotation Tool to Build Large Dependency Tree-Tagged Corpus", *Korean Society for Language and Information ACL*, 2007, pp. 385-396.
- [22] Sapkota, K., Aldea, A., Younas, M., Duce, D.A. and Banares-Alcantara, R. "Semantic-ART: a framework for semantic annotation of regulatory text", *Proceedings of the fourth workshop on exploiting semantic annotations in information retrieval ESAIR*, 2011, pp. 23-24.
- [23] Albukhitan, S. and Helmy, T. "Automatic Ontology-Based Annotation of Food, Nutrition and Health Arabic Web Content", *In Proceedings of the 4th International Conference on Ambient Systems, Networks and Technologies, ANT*, 2013, pp. 461-469.
- [24] Barkschat, K. "Semantic Information Extraction on Domain Specific Data Sheets", *In: Presutti V., d' Amato C., Gandon F., d' Aquin M., Staab S., Tordai A. (eds), The Semantic Web: Trends and Challenges, ESWC, LNCS*, Springer, Vol. 8465, 2014, pp. 864-873.
- [25] Ekuobase, G. O. and Ebietomere, E. P. "Ontology for Nigerian Case Laws", *African Journal of Computing & ICT*, Vol 6, Number 2, 2013, pp. 12-18.
- [26] Bird, S., Klein, E. and Loper, E. "Natural Language Processing with Python", O'Reilly Media Inc, 2009.
- [27] Saggion, H. "Creating Summarization Systems with SUMMA", *LREC*, 2014, pp. 3-7.
- [28] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., and et al. "Developing Language Processing Components with GATE Version 8 – a User Guide", Department of Computer Science, University of Sheffield, UK, 2016.
- [29] Manning, C. and Schütze, H. "Foundations of Statistical Natural Language Processing", MIT press, Cambridge, MA, 1999.