Scientometrics (2022) 127:5459–5473
https://doi.org/10.1007/s11192-022-04280-2

TeknoAssistant : a domain specific tech mining approach for technical problem-solving support

Gaizka Garechana[1] · Rosa Río-Belver[1] · Enara Zarrabeitia[1] · Izaskun Alvarez-Meaza[1]

Received: 1 April 2021 / Accepted: 19 January 2022 / Published online: 5 February 2022

Abstract

This paper presents TeknoAssistant, a domain-specific tech mining method for building a problem–solution conceptual network aimed at helping technicians from a particular field to find alternative tools and pathways to implement when confronted with a problem. We evaluate our approach using Natural Language Processing field, and propose a 2-g text mining process adapted for analyzing scientific publications. We rely on a combination of custom indicators with Stanford OpenIE SAO extractor to build a Bernoulli Naïve Bayes classifier which is trained by using domain-specific vocabulary provided by the TeknoAssistant user. The 2-g contained in the abstracts of a scientific publication dataset are classified in either "problem", "solution" or "none" categories, and a problem–solution network is built, based on the co-occurrence of problems and solutions in the abstracts. We propose a combination of clustering technique, visualization and Social Network Analysis indicators for guiding a hypothetical user in a domain-specific problem solving process.

Keywords TeknoAssistant · Text mining · SAO · Naive Bayes · NLP · Natural language processing

✉ Gaizka Garechana
gaizka.garechana@ehu.eus

Rosa Río-Belver
rosamaria.rio@ehu.eus

Enara Zarrabeitia
enara.zarrabeitia@ehu.eus

Izaskun Alvarez-Meaza
izaskun.alvarez@ehu.eus

[1] Department of Business Management, University of the Basque Country (UPV/EHU), Bilbao, Spain

Introduction

The conception of this paper traces back to our research team's diagnostic of the necessities of organizations with whom we routinely work. Our principal knowledge-transfer activities are focused on a set of tech mining solutions (Porter & Cunningham, 2005) aimed at enhancing the capabilities of technology watch systems in organizations, and the incorporation of the TeknoAssistant represents our answer to a recurrent demand from our collaborators.

The TeknoAssistant stands at the intersection of innovation and technology watch processes (Calof & Sewdass, 2020), helping to merge certain innovation processes with external knowledge data. The incorporation of external knowledge to innovation processes is a central point of open innovation (Chesbrough, 2008) and a relevant input among many, whether related with the performance of the technology watch system or not, that condition the success of the firm's innovation efforts (Nemutanzhela & Iyamu, 2011). The incorporation of external knowledge into decision-making should be facilitated by a smooth-functioning technology watch system, nonetheless, eliminating infoxication remains a challenge. Our solution helps to focus on an input that the aforementioned firms considered important: exact information about technological solutions that are being implemented outside the firm, and then applying this information to the problem solving processes at the firm.

There is a need for an approach which provides curated external information to the problem solving processes inherent to reengineering and design work. Given the lexical biases that are found among different technical fields and inventors (Cascini & Zini, 2008) in descriptions of problems and solutions, we detect a gap in approaches that can combine expert vocabulary from a particular field with the generation of text mining products, while at the same time producing a bespoke solution for a technology watch system.

The approach we describe in this paper is a domain-specific, semi-automated method for extracting and summarizing the scientific problem-solving information in order to feed the technology decision making processes. A significant amount of global scientific research becomes codified in scientific papers, and conveniently stored in scientific databases, forming an easily-accessible, validated and accurate knowledge deposit from which technicians, particularly those working in science intensive fields, can garner valuable clues for problem-solving, both in new product development tasks or when solving technical problems in their respective fields.

The problem lies in activating such decision-making information: The nuts and bolts forming the functional internals of a collection of scientific works are not typically available in a schematic form that would allow, for example, rapid retrieval of a set of solutions applicable to a problem. The authors of this paper are in close contact with decision makers in several technology fields and have often met such necessity, sometimes it is necessary to obtain a bird's-eye view of the latest methods, components or materials that are being put into practice in order to help solve a certain problem or phenomenon that is of interest to the technicians in a given moment. We would like to state that our proposal also obeys the principle of public service (the authors are currently employed by a public university), prompting us to choose a license-free tool to build our TeknoAssistant, our intention being to offer SMEs and other resource-constrained organizations both the impressive possibilities of Python programming language and affordability.

Literature review

Fast, automated analysis of large amounts of textual information falls into the realm of text-mining techniques. This analytical field lies in the conjunction of computer sciences, statistics and information sciences, and aims at the development of quantitative methodologies, among other things, for extracting, retrieving, summarizing and categorizing textual information (Dang & Ahmad, 2013; Sharma & Srivastava, 2016). The bag-of-words is probably the simplest and most frequently used approach when dealing with large amounts of textual information. This type of approach disregards the order of the words in the text and focuses instead on the relative frequency of occurrence of the words in the text. General data cleaning steps and particular text treatment steps (sentence splitting, tokenization, part of speech tagging, parsing, stemming and lemmatization, among others) are required in order to separate the informative words from the connectors and other syntactic items that do not convey significant conceptual meaning, as well as for merging the declensions of the words (Jo, 2019). It should be noted that when analyzing textual information about technology components or methods, it is convenient to include bigrams and/or trigrams in the analysis, since multi-word combinations can be indicative of problems and solutions (Heffernan & Teufel, 2018). Multi-word based text mining results also improve the interpretation of text-mining analysis results, especially when dealing with complex topics such as technology solutions (Xu et al., 2021) or clinical risk factors (Sabra & Sabeeh, 2020), consequently, an approach based on n-grams seems to be the best choice when dealing with technically complex and context-dependent subjects.

While sharing an important part of the text mining workflow with the previously described approaches, semantic techniques are in a league of their own. These techniques aim at providing an "understanding" dimension to the automated analysis of text, so the role each word plays in the sentence can be incorporated into the analysis. Regarding the word's role, verbal exposure of problem–solution structures is particularly useful for tech mining purposes: the automated extraction of a set of problems from a text corpus, accompanied by its associated solutions, can be a valuable input for many problem-solving tasks in manufacturing and engineering. The TRIZ methodology (Altshuller, 1984) relies on the analysis of information to detect "unresolved contradictions" that form the basis of its inventive problem-solving process. Verbitsky (2004) coined the expression "semantic TRIZ" and linked Subject-Action-Object linguistic structures to the automated detection of such unresolved contradictions. The identification of such Subject-Action-Object (SAO) structures in textual information is a relevant goal of semantic analysis, since these structures provide a linkage between two or more conceptual items that are present in the SAO based sentence, either acting as a subject that performs an action or as an object on which the action is performed (Yang et al., 2017a, 2017b; Zhang et al., 2014, 2021). This type of information can be determinant for complementing expert judgment in the analysis of a scientific or technological field, providing a desirable quantitative support to decision making in this field (Chen et al., 2015). In this work we propose a methodology for detecting and analyzing the problem–solution structures present in a scientific field, combining the features of SAO analysis with classic bag-of-words indicators applied to a 2-g scientific information database.

There are several precedents in the use of syntactical indicators as a technique for discarding irrelevant informational items and for analyzing the relationships between concepts present in scientific and technological information. Yoon and Kim (2012) propose a syntactic-dependence based system for assisting experts in identifying technology trends in technology forecasting exercises. Their method automatically extracts properties and functions of inventions from the sentences contained in patent data, properties refer to methods applied for such purposes, whereas functions relate to the usage. This information is then fed to a TRIZ-like system that will give quantitative assistance to the experts trying to forecast the evolution of technology in a given area.

SAO objects are an interesting syntactic structure for science and technology (S&T) characterization. They provide threefold information regarding "what is exerting an action on what", and depending on the nature of that action (the type of verb linking the subject and the object) different types of SAO objects can be defined (Choi et al., 2013), leading to the definition of ontologies for guiding decision making in S&T. Looking at the type of SAO structures detected, the verbal structure formed by action-object pairs could be classified in a particular syntactic category that could be assigned to a specific part of an invention, such as a material type, a product or a technology (Choi et al., 2012). The relevance of SAO objects can be determined by a scoring system based on keyword dictionaries, in order to extract the objects with the highest potential to solve the research question being addressed, as shown by Cascini et al. (2004), while Yang et al., (2017a, b) propose using the SAO objects for first identifying technology requirements and then building a "relevance indicator" that will identify the core technological components aimed at fulfilling such requirements. There are several potential applications of SAO objects in science/technology decision making tools: Wang et al. (2015) propose a method for building technology roadmaps using networks formed by SAO objects, looking for the identification of the components present in a scientific information database on the one hand, and on the other, extracting the answers to relevant technology questions by analyzing the relationships in the SAO network. In addition to roadmapping and forecasting, SAO objects present in patent data have also been used for building a TRIZ methodology, concluding that the core concepts forming an invention are properly captured by these objects (Park et al., 2013). There is also ample literature dealing with SAO object applications to characterize technology trends. SAO objects present in patent text could be used to calculate the similarity between patents (Yoon & Kim, 2011a), while Yoon and Kim (2011b) use a similar approach to detect potential niches for radical innovations. Some authors establish an SAO object taxonomy based on the nature of the relationship between subject and object: partitive SAOs correspond to an inclusion relationship whereas attribute SAOs correspond to sentences in which the subject modifies the object to some extent (Wang et al., 2017). The relationship network formed by SAO objects present in a patent dataset has also been successfully analyzed using social network analysis tools to identify technology trends (Choi et al., 2011).

Finally, It should be noted that SAO structures are far from infallible and are subject to the same limitations of any Natural Language Processing (NLP) technique, especially when confronted with the ambiguities of human language and the fuzzy semantics of complex sentences (Abbas et al., 2014).

Methodology

Data sample

In order to interpret the results of the methodology adequately, we consider the expertise of the authors in NLP field and then build a sample formed by scientific publications in NLP, by running the query (TITLE ("natural language processing") OR ABS ("natural language processing")) AND (EXCLUDE (PUBYEAR, 2021)) on Scopus database. Among these, the contributions with empty "Abstract" or "Author Keywords" fields are discarded, since our methodology requires that information. This produces a sample consisting of 15,764 publications for the interval 1968–2020.

Text mining process

Our approach is based in a 2-g analysis, considering that both problems and solutions in a particular field are frequently expressed using tuples formed by 2 or more terms. The "Title", "Abstract" and "Author Keywords" fields are processed to form database A according to this workflow (upper branch in Fig. 1):

1. Stanza analysis package is used (Qi et al., 2020) for sentence splitting and lemmatization. This neural network based Python package is especially suited for text analytic purposes in several languages.
2. We built Python code for data cleaning and relied on the NLTK platform (Bird et al., 2009) for the building of a lemmatized 2-g database (database A) containing the textual information present in "Title" and "Abstract" and "Author keyword" fields. We adapt the code to detect acronyms in the "Author Keyword" field, so we can match an acronym with a 2-g containing it.

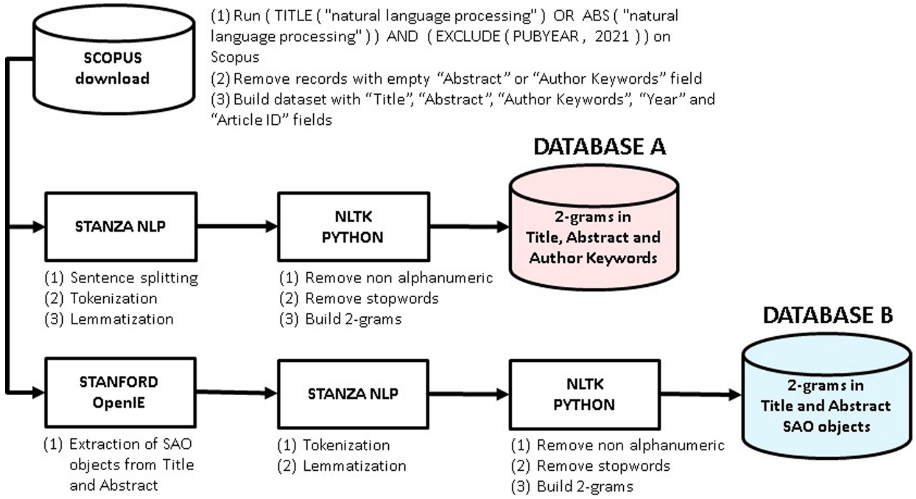

SCOPUS download
(1) Run (TITLE ("natural language processing") OR ABS ("natural language processing")) AND (EXCLUDE (PUBYEAR , 2021)) on Scopus
(2) Remove records with empty "Abstract" or "Author Keywords" field
(3) Build dataset with "Title", "Abstract", "Author Keywords", "Year" and "Article ID" fields
DATABASE A
STANZA NLP
(1) Sentence splitting
(2) Tokenization
(3) Lemmatization
NLTK PYTHON
(1) Remove non alphanumeric
(2) Remove stopwords
(3) Build 2-grams
2-grams in Title, Abstract and Author Keywords
DATABASE B
STANFORD OpenIE
(1) Extraction of SAO objects from Title and Abstract
STANZA NLP
(1) Tokenization
(2) Lemmatization
NLTK PYTHON
(1) Remove non alphanumeric
(2) Remove stopwords
(3) Build 2-grams
2-grams in Title and Abstract SAO objects

Fig. 1 Workflow of text mining process

3. The process finishes with the building of database A, a database containing the 2-g detected in title, abstract and keyword fields.

Separately (database B) we use the Python wrapper (Remy, 2020) for the Stanford SAO extractor Open Information Extraction (OpenIE) (Manning et al., 2014) to extract the SAO objects present in "Title" and "Abstract" fields. In order to have the same, homogenized versions of the 2-g in this database (database B), the lemmatization, cleaning and 2-g building steps (when necessary) of SAO objects have been performed by adapting the code used for building database A (see Fig. 1).

This process builds two separate 2-g databases formed by standardized tuples. The presence of a 2-g from database A in database B can be used as an input data in order to determine if such 2-g can be labeled as the "subject" or "object" forming part of a SAO structure, according to the OpenIE algorithm.

A description of the main text mining steps explained in this section is provided in Table 1, indicating the source and the URL directing to the website of each main tool, where indications for installation are provided.

Domain-specific approach for problem/solution identification

TeknoAssistant is an approach for building domain-specific problem–solution structures that can assist technicians involved in new product development or confronted with technical problems in their respective fields. We propose an approach that complements the SAO object identification of OpenIE algorithm with a set of features that can be calculated from the information available in databases A and B. TeknoAssistant would classify the standardized 2-g present in scientific publications as a "problem" or "solution" by using a multivariate classifier trained with a set of domain-specific examples. It should be noted that in the application area (NLP) chosen for this study, "problems" are often expressed as data sources to exploit, whereas "solutions" are often algorithms and other analysis tools.

Figure 2 shows the vector of variables that will feed TeknoAssistant, which is built using the data from three data sources, namely, database A and B described in the previous section and the set of domain-specific examples of problems and solutions provided by the user. The domain specificity of TeknoAssistant depends on this input, so users should be given instructions on how to succinctly express the denomination of the typical problems and solutions in their field in order to feed TeknoAssistant, such as "axis friction" and "solid lubricant", for example.

For each of the 2-g in the abstracts from database A, the following variables will be defined, in addition to the dependent variable:

4. X_1: Presence of the 2-g in the field "Author Keywords" (binary).
5. X_2: Presence of the 2-g in the field "Title" (binary).
6. X_3: The 2-g is present in the field "Abstract" twice or more (binary).
7. X_4: The 2-g is identified as an SAO subject by the OpenIE extractor in a sentence of the field "Abstract" (binary).
8. X_5: The 2-g is identified as an SAO object by the OpenIE extractor in a sentence of the field "Abstract" (binary).
9. X_6: The 2-g is identified as an SAO subject by the OpenIE extractor in a sentence of the field "Title" (binary).

Table 1 Main steps explained in this section, with the corresponding description and source

Step of the process	Purpose	Method details
Sentence splitting, tokenization, lemmatization	Obtain the homogenized lemma of the words present in data	NLP process implemented in Stanza Python package (Qi et al., 2020) https://stanfordnlp.github.io/stanza/
Special character and stopword removal, building of 2-g	Eliminate non-informative items from the data, transform data into a 2-g database	Libraries at NLTK Python package (Bird et al., 2009) https://www.nltk.org/
Extraction of SAO objects from title and abstract	Identify the SAO models detected using a different methodology	Stanford OpenIE SAO extractor implemented in Python (Remy, 2020) https://nlp.stanford.edu/software/openie.html

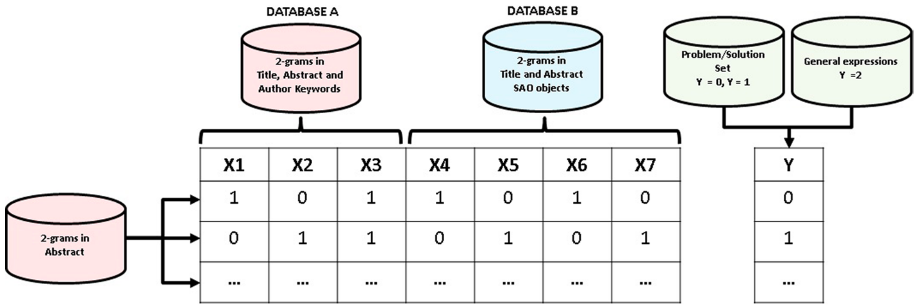
10. X_7: The 2-g is identified as an SAO object by the OpenIE extractor in a sentence of the field “Title” (binary).
11. Y: The 2-g is labeled as a problem ($Y = 0$), solution/tool ($Y = 1$) or none ($Y = 2$), using the domain-specific problem/solution set. The “none” label is assigned using a list of general expressions in scientific abstracts such as “extant literature” or “provide opinion” that can be complemented by the user with terms too vague to indicate a specific problem or solution of interest, such as “machine learning”, in this case. The $Y = 3$ value is assigned if the 2-g from the abstract does not match with any of the expressions present in the domain-specific problem/solution set or in the list of general expressions.

This process generates a vector formed by 7 independent variables for each of the 2-g present in the abstracts, some of which will match with the labeled 2-g from the domain-specific problem/solution set. We use these to train and test a Bernoulli Naïve Bayes (NB) classifier implemented in Scikit-learn machine learning library for Python (Pedregosa et al., 2011). The resulting model will be used to label the unmatched 2-g ($Y = 3$) from the abstracts as “problem”, “solution” or “none”. The NB classifier occasionally classifies the same 2-g under different categories, in which case the most frequent category is used to label the 2-g.

We end this point of the methodology having produced a set of problems and solutions as a result of combining the predictions of the NB model with the items provided by the user (training and test sample). We decided to establish the relationship between problems and solutions by using the co-occurrence of both items in the same abstract, it seems reasonable to expect that a recurrent (in subsequent analysis steps single occurrences are going to be excluded) co-occurrence of a problem–solution pair in the abstract field can be evidence of a relationship between the two.

Clusterization, mapping and social network analysis (SNA) indicators

In order to present the results of this approach to the user, the clustering and mapping of the problem–solution network is proposed. The clusterization algorithm and layout technique implemented in VOSviewer software (Van Eck & Waltman, 2010) are used for generating the visualization of the network, eliminating the problem–solution pairs that were


DATABASE A
2-grams in Title, Abstract and Author Keywords
DATABASE B
2-grams in Title and Abstract SAO objects
Problem/Solution Set Y = 0, Y = 1
General expressions Y =2
2-grams in Abstract

X1	X2	X3	X4	X5	X6	X7	Y
1	0	1	1	0	1	0	0
0	1	1	0	1	0	1	1
...	...	...	...	...	...	...	...

Fig. 2 Building the variable vector for each 2-g present in the abstracts

present in less than 5 papers, as well as problems or solutions that did not have a counterpart in the same abstract, for example, a 2-g labeled as “problem” with no “solution” 2-g present in that same abstract.

We aim at identifying key problems and solutions that could be of interest because they act as a link between different clusters of the network. With this purpose in mind, we use a Social Network Analysis (SNA) indicator, namely, the betweenness centrality. Typically, high-degree nodes in a network also have high values in betweenness centrality, so we propose to focus the analysis on low degree nodes that “punch above their weight” in the betweenness centrality indicator. This task is achieved by excluding the top 15 nodes with the highest degree and studying the remaining nodes with the highest betweenness centrality. This indicator is calculated using Gephi SNA software (Bastian et al., 2009).

A description of the main steps explained in this section is provided in Table 2, indicating the URL directing to the website of each tool, where indications for installation and description of algorithms are provided.

In addition to this, our approach allows the user to perform manual checking of intermediate solutions or innovation pathways. The network can be useful for guiding the user through a set of connected problem–solution structures that may suggest alternative solutions to the problem in hand.

A fully automated implementation of TeknoAssistant would allow the user to perform such explorations via a software interface, thus forming an ancillary resource for solving problems in product design or manufacturing, as shown in Fig. 3.

Results and discussion

The number of scientific publications in NLP, according to our query, has experienced an exponential growth since the first record in year 1968, as shown in Fig. 4. This result is perfectly coherent with the natural growth in global scientific activity and the interest raised by the Artificial Intelligence NLP applications, in addition to the increased availability of textual information sources provided by the Internet, among other factors.

After removing the 2-g derived from “natural language processing” and the acronym “NLP” from Database A, we end with 1,763,742 2-g extracted from the abstracts of the publications. We decided that this removal was necessary since these 2-g were present in almost all the publications in the sample, due to our query-based data retrieval. In addition to this, we feed TeknoAssistant with 53 domain-specific problems, 104 solutions and 392

Table 2 Main steps explained in this section, with the corresponding description and source

Step of the process	Purpose	Method details
Mapping and clusterization of data	Identifying closely related problem/solution relationships and building a human-readable visualization that can be directly fed into technical decision making	Clusterization and layout algorithm implemented in VOSviewer software https://www.vosviewer.com/ Documentation of VOSviewer https://www.vosviewer.com/documentation/Manual_VOSviewer_1.6.8.pdf
SNA indicators	Centrality and betweenness centrality indicators are calculated in order to interpret the role that concepts are playing in the problem/solution network	The SNA indicators used in this study are part of the common knowledge in graph theory, and have been calculated using Gephi https://gephi.org/

common expressions. The result of running the steps described in section "5" are given in Table 3.

The labeled data consists of 44,235 2-g and we build the training and test data by shuffling these labeled 2-g and splitting the data, 80% of the data (35,388) is set aside for the training data and the remaining 20% (8847) makes the test data. The NB classifier correctly classifies 5827 2-g from the test data, thus returning 66% accuracy, which we deem to be acceptable considering the complexity of the task at hand, since the semantics that differentiate problems from solutions can be quite subtle.

The trained NB model is used to predict the unlabeled data, so all the 2-g present in the 15,764 abstracts are classified in one of the categories "problem", "solution" or "none". A new dataset is built containing all the problem–solution pairs detected, using the co-occurrence of both items in an abstract as evidence of relationship, as explained in the methodology. We obtained 34,746 problem–solution pairs that are exported to VOSviewer software in order to generate the clusterization and mapping of the resultant undirected problem–solution network, shown in Fig. 5.

This network is dominated by two closely related solutions in NLP fields, namely neural networks (NN) and Deep Learning (DL) algorithms, while several well-known NLP problems such as automatic translation, sentiment analysis, information retrieval and question-answering systems, among others, are clearly present on the right side of Fig. 5. Research in this field is apparently dominated by two large, related, machine learning tools that offer a wide range of variations and complements, while the problems being addressed are far more heterogeneous. The flexibility of the NN and its variants makes them a "one-size-fits-all" solution for solving several NLP problems, this being a particular feature of this field. A deeper look into the data using VOSviewer shows that NN and DL are both connected to all the prominent problems in the problem–solution network. The clusterization algorithm has adequately captured the specialization of Long Short Term Memory (LSTM) algorithms in speech recognition, and it is interesting to note that information retrieval and sentiment analysis problems share a certain amount of strong links with some popular tools or "solutions" in NLP, namely decision trees, word embedding and topic modeling. This impression is reinforced when analyzing the betweenness centrality of solutions that could be of interest because they act as a link between different clusters of the network. Three techniques (topic modeling, decision tree and random forest) are in the top 10 betweeners corrected by node degree (see 3), linking together information retrieval and sentiment analysis problems.

As an example, here are some interesting clues, from a problem-solving point of view, that a hypothetical user can extract from the visual analysis of the network. Commenting on Fig. 6 from left to right, we observe that "social media" represents a data source that may offer suitable solutions to solve sentiment analysis and information retrieval problems, in addition to this, the connection with the "adverse drug" node informs us that there are several applications of social media data to detect adverse reactions to drugs, among others. Social media data is identified as a transversal data source for solving several NLP issues.

The second figure, on the other side, is telling us that the application of LSTM in time series data can open the door to solving problems in the speech recognition field, while the third visualization informs us about the versatility of logistic regression for dealing with certain aspects of the most relevant problems in NLP field. It should be noted that in areas strongly dominated by a reduced number of really popular solutions (such as NN and DL in NLP) the detection of less-known versatile tools that show application in several problems can be extremely valuable to technicians who have become mentally blocked within a herd mentality.

Conclusions

In this paper we have explained the internals of TeknoAssistant, a tech temining approach aimed at supporting technical problem-solving by providing a problem–solution network to head technicians, as well as valuable clues pointing at potential solution pathways being

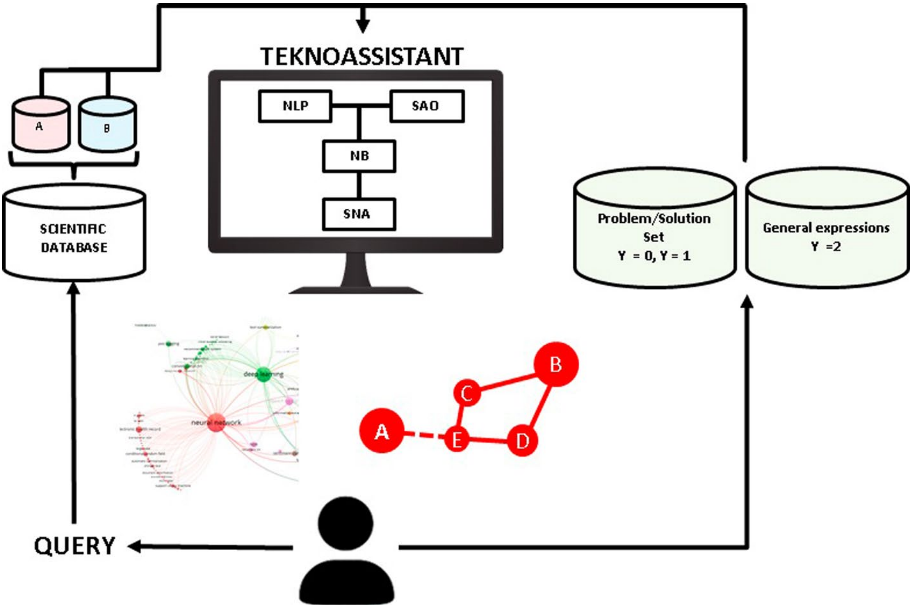


Fig. 3 Depiction of the implementation of TeknoAssistant in a workplace

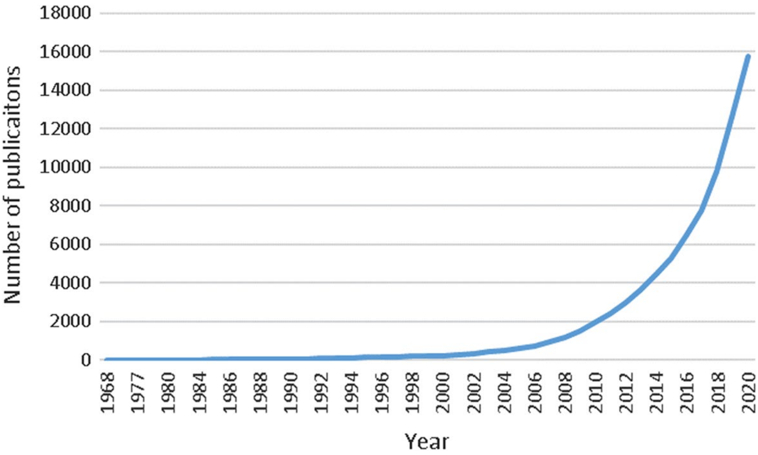


Fig. 4 Evolution of the number of publications since 1968

Table 3 Result of labeling the 2-g in Database A

Label	Number of 2-g
Problem	9311
Solution	7538
None	27,386
Unlabeled	1,719,507

ignored. The text mining process followed for building homogenous 2-g databases and the subsequent matching of data is described, followed by the criterion for building of a set of seven independent variables with which to label every 2-g in the data either as "problem", "solution" or "none". A Bernoulli NB classifier has been trained for this labeling task and problem–solution pairs have been defined by pairing the labeled 2-g that co-occur in the abstracts. The resulting network has been analyzed and its potential for the purposes of TeknoAssistant has been addressed by combining the visualization analysis with SNA indicators.

TeknoAssistant shows that NLP solutions are clearly dominated by two broad categories of machine learning algorithms, namely NN and DL which, at the same time, share strong methodological principles. The performance and flexibility of these techniques makes them a multipurpose tool with which to address multiple NLP problems. The problems, on the other hand, show a well-defined heterogeneity and valuable insight into the nature of some of the solutions that can be obtained by exploring the network, for example, the interface between information retrieval and sentiment analysis problems shows that there are several tools which can be applied to solve problems in both areas. This can provide major pointers for technicians that are stuck on a particular problem in one of these areas, informing them about tools that might be popular in the adjacent area but are being dismissed at this particular time. The betweenness centrality indicator has also produced alerts regarding some of these tools, so there is evidence pointing at the validity of this indicator to reinforce the visualization analysis of TeknoAssistant.

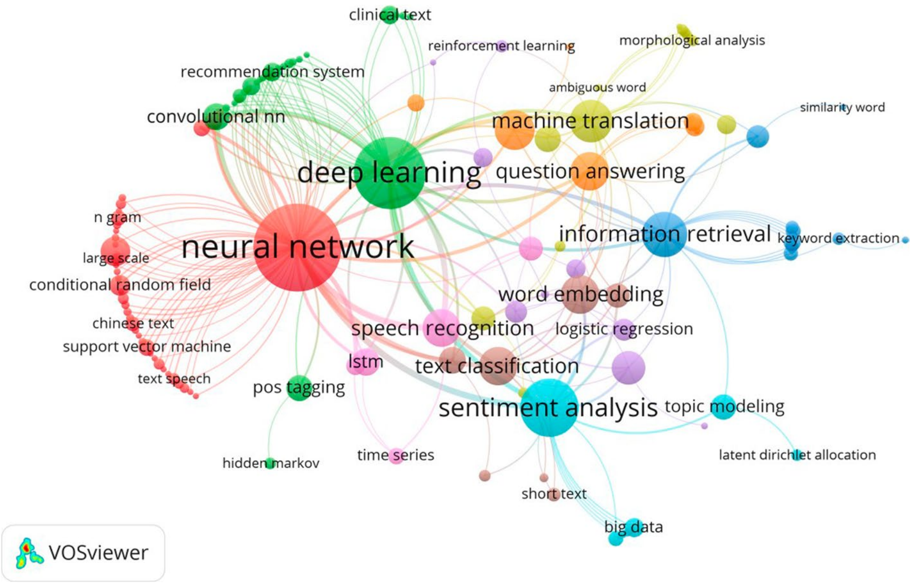


Fig. 5 Problem–solution network of NLP, relations based on less than 5 publications have been eliminated. Size of nodes reflects the number of publications in which each concept occurs

Deeper navigation into the problem–solution network is also encouraging: transversal data sources that could open the way to solving problems in related areas are detected, as well as clues for complementing some techniques. The technique also successfully manages to detect valuable solutions that may not be at the forefront of cutting edge, providing the technician with a useful signaling approach for expanding the set of tools that is currently being deployed to solve a problem.

Regarding the limitations of the study, the accuracy of the NB classifier is lower than desirable, and this is noticeable particularly in the "problem" vs "solution" classification. In our method, 4 variables out of 7 rely on the Stanford OpenIE SAO object extractor accuracy, which may not be fully suitable for our purpose. In any case, the authors are aware that the performance of the classification task leaves room for improvement at this stage of development of TeknoAssistant. We also found that expanding the domain specific problem–solution set did not lead to improved accuracy, quite the contrary, our methodology showed that the best results were obtained by using a reduced set of

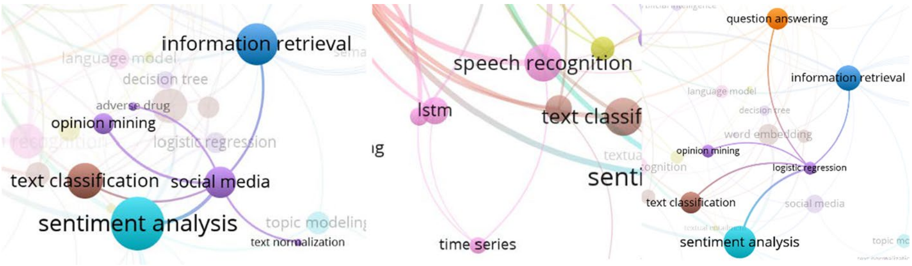


Fig. 6 Screenshot of several clues that a hypothetical user can extract from the problem–solution network

well-established, popular problems and solutions for training the model. Another aspect that could be a determinant in explaining the relatively low accuracy lies in the length of the n-grams that would be necessary in order to capture the full expression that alludes to a certain problem or solution, such as “General Regression Neural Network”. The expansion of TeknoAssistant to include the analysis of 3 or 4-g would probably require a trade-off between computational efficiency and the accuracy of the final results, and would also depend on the particularities of the vocabulary in the field being analyzed.

Finally, it is worth noting that TeknoAssistant is meant to be a tech mining method aimed at guiding a more-or-less obfuscated technician through a brainstorming-like process in which important clues and insights should be revealed. The fine tuning of such a system is only to be achieved by continuous feedback loops with the decision makers in each particular field of application. In addition to this, the validation of the methodology behind the TeknoAssistant also depends on such interaction with technology experts.

References

Abbas, A., Zhang, L., & Khan, S. U. (2014, June 1). A literature review on the state-of-the-art in patent analysis. *World Patent Information*. Elsevier Ltd. https://doi.org/10.1016/j.wpi.2013.12.006

Altshuller, G. S. (1984). *Creativity As an Exact Science*.

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. In *AAAI Conference on Weblogs and Social Media.*

Bird, S., Loper, E., & Klein, E. (2009). Natural Language Processing with Python. O’Reilly Media Inc.

Calof, J., & Sewdass, N. (2020). On the relationship between competitive intelligence and innovation. *Journal of Intelligence Studies in Business, 10*(2), 32–43. https://doi.org/10.37380/JISIB.V10I2.583

Cascini, G., & Zini, M. (2008). Measuring patent similarity by comparing inventions functional trees. *IFIP International Federation for Information Processing, 277*, 31–42. https://doi.org/10.1007/978-0-387-09697-1_3

Cascini, G., Fantechi, A., & Spinicci, E. (2004). Natural language processing of patents and technical documentation. *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3163*, 508–520. https://doi.org/10.1007/978-3-540-28640-0_48

Chen, H., Zhang, G., Zhu, D., & Lu, J. (2015). A patent time series processing component for technology intelligence by trend identification functionality. *Neural Computing and Applications, 26*(2), 345–353. https://doi.org/10.1007/s00521-014-1616-y

Chesbrough, H. (2008). Open Innovation: A new paradigm for understanding industrial innovation. In *Open Innovation: Researching a New Paradigm* (pp. 1–15). Oxford University Press. https://books.google.com/books?hl=es&lr=&id=RdcSDAAAQBAJ&oi=fnd&pg=PA1&dq=external+knowledge+open+innovation&ots=kRQb30N8D9&sig=EMrZbwF3eUcdYKpKzBi-wdRPB_A. Accessed 9 October 2021.

Choi, S., Yoon, J., Kim, K., Lee, J. Y., & Kim, C.-H. (2011). SAO network analysis of patents for technology trends identification: A case study of polymer electrolyte membrane technology in proton exchange membrane fuel cells. *Scientometrics, 88*(3), 863–883. https://doi.org/10.1007/S11192-011-0420-Z

Choi, S., Park, H., Kang, D., Lee, J. Y., & Kim, K. (2012). An SAO-based text mining approach to building a technology tree for technology planning. *Expert Systems with Applications, 39*(13), 11443–11455. https://doi.org/10.1016/j.eswa.2012.04.014

Choi, S., Kim, H., Yoon, J., Kim, K., & Lee, J. Y. (2013). An SAO-based text-mining approach for technology roadmapping using patent information. *R&D Management, 43*(1), 52–74. https://doi.org/10.1111/j.1467-9310.2012.00702.x

Dang, S., & Ahmad, P. H. (2013). *A Review of Text Mining Techniques Associated with Various Application Areas*. *International Journal of Science and Research* (Vol. 4). www.ijsr.net. Accessed 11 January 2021

Heffernan, K., & Teufel, S. (2018). Identifying problems and solutions in scientific text. *Scientometrics, 116*(2), 1367–1382. https://doi.org/10.1007/s11192-018-2718-6

Jo, T. (2019). *Text mining*. (Janusz Kacprzyk, Ed.)*Studies in Big Data*. Springer International Publishing AG. https://link.springer.com/content/pdf/https://doi.org/10.1007/978-3-319-91815-0.pdf. Accessed 8 October 2021

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60).

Nemutanzhela, P., & Iyamu, T. (2011). The impact of competitive intelligence on products and services innovation in organizations. *International Journal of Advanced Computer Science and Applications, 2*(11), 38–44.

Park, H., Ree, J. J., & Kim, K. (2013). Identification of promising patents for technology transfers using TRIZ evolution trends. *Expert Systems with Applications, 40*(2), 736–743. https://doi.org/10.1016/j.eswa.2012.08.008

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net. Accessed 26 March 2021

Porter, A. L., & Cunningham, S. W. (2005). *Tech mining: Exploiting new technologies for competitive advantage*. Wiley-Interscience.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. Association for Computational Linguistics (ACL) System Demonstrations.

Remy, P. (2020). Python wrapper for Stanford OpenIE. GitHub.

Sabra, S., & Sabeeh, V. (2020). A Comparative Study of N-gram and Skip-gram for Clinical Concepts Extraction. *Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020*, 807–812. https://doi.org/10.1109/CSCI51800.2020.00151

Sharma, S., & Srivastava, S. (2016). Review on text mining algorithms. *International Journal of Computer Applications, 134*(8), 39–43.

Van Eck, N. J., & Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics, 84*(2), 523–538.

Verbitsky, M. (2004). Semantic TRIZ. *triz-journal.com*. https://pdfs.semanticscholar.org/a3fe/e18cae12fb8a57a966442fbf40e387d0fc98.pdf

Wang, X., Qiu, P., Zhu, D., Mitkova, L., Lei, M., & Porter, A. L. (2015). Identification of technology development trends based on subject-action-object analysis: The case of dye-sensitized solar cells. *Technological Forecasting and Social Change, 98*, 24–46. https://doi.org/10.1016/j.techfore.2015.05.014

Wang, X., Ma, P., Huang, Y., Guo, J., Zhu, D., Porter, A. L., & Wang, Z. (2017). Combining SAO semantic analysis and morphology analysis to identify technology opportunities. *Scientometrics, 111*(1), 3–24. https://doi.org/10.1007/s11192-017-2260-y

Xu, S., Hao, L., Yang, G., Lu, K., & An, X. (2021). A topic models based framework for detecting and forecasting emerging technologies. *Technological Forecasting and Social Change, 162*, 120366. https://doi.org/10.1016/J.TECHFORE.2020.120366

Yang, C., Zhu, D., & Wang, X. (2017a). SAO semantic information identification for text mining. *International Journal of Computational Intelligence Systems, 10*(1), 593–604. https://doi.org/10.2991/ijcis.2017.10.1.40

Yang, C., Zhu, D., Wang, X., Zhang, Y., Zhang, G., & Lu, J. (2017b). Requirement-oriented core technological components' identification based on SAO analysis. *Scientometrics, 112*(3), 1229–1248. https://doi.org/10.1007/s11192-017-2444-5

Yoon, J., & Kim, K. (2011). Identifying rapidly evolving technological trends for R&D planning using SAO-based semantic patent networks. *Scientometrics., 88*(1), 213–228. https://doi.org/10.1007/S11192-011-0383-0

Yoon, J., & Kim, K. (2011b). Detecting signals of new technological opportunities using semantic patent analysis and outlier detection. *Scientometrics, 90*(2), 445–461. https://doi.org/10.1007/S11192-011-0543-2

Yoon, J., & Kim, K. (2012). TrendPerceptor: A property-function based technology intelligence system for identifying technology trends from patents. *Expert Systems with Applications, 39*(3), 2927–2938. https://doi.org/10.1016/j.eswa.2011.08.154

Zhang, Y., Zhou, X., Porter, A. L., & Vicente Gomila, J. M. (2014). How to combine term clumping and technology roadmapping for newly emerging science & technology competitive intelligence: "problem & solution" pattern based semantic TRIZ tool and case study. *Scientometrics, 101*(2), 1375–1389. https://doi.org/10.1007/s11192-014-1262-2

Zhang, Y., Wu, M., Hu, Z., Ward, R., Zhang, X., & Porter, A. (2021). Profiling and predicting the problem-solving patterns in China's research systems: A methodology of intelligent bibliometrics and empirical insights. *Quantitative Science Studies, 2*(1), 409–432. https://doi.org/10.1162/QSS_A_00100