Cahiers
#Costech

> Eglantine Schmitt

Epistemic Regimes in Natural Language Processing

> #Numéro 3
> Philosophie des techniques
> Notes de recherche
> EPIN - Ecritures, Pratiques et Interactions Numériques (Costech-UTC)
> Humanités numériques plurielles - > Mégadonnées - > Méthodologie de recherche - > Philosophie

Citer cet article

Schmitt, Eglantine. "Epistemic Regimes in Natural Language Processing.", 7 juillet 2019, *Cahiers Costech*, numéro 3.
URL https://costech.utc.fr/CahiersCostech/article82.html

With the exponential development of individual and infrastructural computer use, many aspects of human sociability and behaviour have been captured through data, including scientific practice. In correlation with the rise of data volume, new computer techniques have been designed and developed while previous one have become more popular than before. Some of them, such as artificial neural networks, need large amounts of data to give relevant results, and high computing power to work. This rise and rebirth of data analysis is what has been named “data sciences”, in which data can be numbers, text, images, or other kinds of digital information. The data science phenomenon, both academic an industrial, currently has a reputation of objectivity and universal effectivity largely spread by media and corporate communication.

From the non-practitioner point of view, data is credited with a primary positivity where truth will emerge from data if the right techniques are used. It is no empiricism in the traditional way because data is not a measure of reality but a fact in itself ; its relationship to the observable world is not built through scientific instruments and local epistemologies as in the natural sciences. From an ontological point of view, there might very well be no reality beyond data. To data scientists, it comes as an absolute given and a starting point that must be trusted for the analysis to work. Data manipulation is a way to learn about the data, but not necessarily about the underlying events of the world that it might represent.

Through a case analysis, I would like to argue that data scientists are very well aware of the constructive nature of data as they witness it in their daily work ; data is *created* by its collection and its technical formatting. Instead of data *sciences,* they would rather describe their practices as data crafts : a set of theoretical knowledges, but also intuition, know-how, experience, and of course technical proficiency, that is relevant to the art of data manipulation. In the lack of a formal science of data, and as they go through the process of data manipulation et visualisation, they manifest a multitude of epistemic regimes, a variety of mental representation of what is knowledge : sometimes, the outcome of their work may appear as solid truth, and shortly afterwards, as strongly constructed by their choices, or even completely made up. Not only are those regimes changing over time, but they are not something that is explicitly formulated : it is my interpretation of how they understand their own work. As a consequence, the idea that those processes of data manipulation might be automated appear not only as highly improbable, but also as not recommendable.

To demonstrate this variety of epistemic regimes in data crafts, I use

the experience of my colleagues and my very own at the company where I work. This company is Proxem, a software publisher which specialises in text analysis and regards itself as a data science player. Among other things, I work there as a business analyst and use this activity as a case study for my ongoing Ph.D. thesis. This case has three specificities which may or may not be typical of data sciences in general :

- text is unstructured data, as opposed to what is usually called “structured data” in the industry, that is tables of quantitative or qualitative variables ;
- data analysis is distributed among several agents of distinct technical and epistemic cultures ;
- the analysis has an industrial focus rather than an academic one, and as such does not seek to establish scientific truths although it deals with scientificity criteria.

Text analysis is made possible by relying on natural language processing (NLP), a domain of computer science and artificial intelligence which focuses on modelling, manipulating and understanding text data. From an historical point of view, NLP used to be a subset of artificial intelligence that relied on formal grammars and morphosyntactic analysis to understand language, with tight links to formal linguistics. Nowadays, it has taken a rather statistical turn where modelling and understanding is no longer required as vast amounts of data and machine learning algorithms yield practical uses and results. NLP and linguistics have taken different paths while computational linguistics continues to exist as a set of tools for linguists (or social scientists) but has not set the design of these techniques as a purpose for itself. In this way, linguistics seems to have a path similar to phonetics which has reborn amongst engineers and computer scientists as speech analytics (Grossetti & Boë 2008).

Over the years, Proxem has found several applications of text analysis that can be monetized. The most frequently sold application is codification automation. In quantitative social sciences and corporate surveys are often included one or more “open questions” that the surveyee can answer not by selecting option(s) in a list (as for closed-ended questions) but with a short text. Each of these is called a verbatim. More recently, pollsters have designed surveys where there are almost no closed-ended questions left ; for example the Net Promoter Score methodology requires only a score which expresses how likely the respondent would recommend something and a justification. More largely, Amazon comments, forum posts, tweets and many other channels are now considered relevant sources of verbatims. By extending the scope of sources, major French corporate

pollsters working with Proxem accept to grow apart from the orthodox methodology of quantitative surveys where the validity of the results relies on the representativeness of the surveyed sample of the population. Since this representativeness cannot be achieved with those new sources of data, they have no other choice but to renounce to them, at least for the construction of the dataset. This representativeness can be challenged by several biases that come from the sampling method, the collection method and the distribution of non-responses (Frippiat & Marquis 2010). Without sampling, it can be assumed but never statistically proven. Corpus linguistics deals with similar issues, since the studied set of texts must constitute a coherent whole (Cailliau & Poudat 2008). Both statistical and linguistic methodologies view the variety of sources and lack of control over them as a diminution of the quality and representativeness of data. From their norms, the rise of “big data” and its data variety is rather a risk than an opportunity ; similarly, division of work and verbatim reuse make it necessary to work on secondary data which is not always documented, and whose connection to public opinion can only be assumed.

Proxem’s customers are pollsters and market research departments who see their verbatims from this methodological culture. When the data is handed to Proxem, it always already has an epistemic value conferred by the customer, and Proxem is not held responsible for the non-representativeness of data. At this step of the process, methodological reservations must be put aside and give way to a data positivism which is the condition of possibility of any further analysis. It is only because data has its own positivity from the customer’s point of view that Proxem can analyse it and make sense of the results. If the input has no epistemic value, the output will not either. In a way, meaning is in the eye of the recipient, for data appreciation may depend on the person, and the individuals that will analyse the data are not the same as those who collected it.

Strickly speaking, data analysis can only begin after data has been imported into the software. For that, it must be formatted and normalized so that it becomes compatible with the software system. Whatever the data contains, it must go through this step which sends it back to its technical and digital nature. Technically speaking, data is a set of entries that share (at least partially) the same variables, so that entries who don’t have “missing values”. Empirically, there are many ways to substantiate this formalism (databases, data format exchanges, tables…) which are more or less compatible between themselves, but to fit in one of them it is the formal condition of possibility of any further data handling. The conversion process is not a purely technical one ; choices have to be made about what can be

merged and what will be visible. The construction of data happens during the data collection but also during this step of formatting. For instance, a frequent question for Proxem engineers is “what is a document ?”. If a survey contains two open ended-questions, should they be merged into one document so that there is one document for each surveyee, or should they be kept apart, hence giving less weight to surveyees who answered only one of the two questions ? If the text data is a conversation between a customer and a hotliner, should each reply be considered a document, or the whole conversation ? Choices must also be made about what a question is : if questions happen to slightly vary over time on a longitudinal study, or because surveyors use slightly different formulations, should they be considered the same questions (so that the answers are stored into the same variables) or kept separate, because some pollsters would consider them not comparable ? When do two questions become distinct ? When the engineer imports data, he does not only write or use pieces of computer code, but also makes choices about the data he will analyse. Those choices, that will ultimately affect the quantification and prioritization of data and ideas, can come from a variety of reasons : the current situation, previous choices made by the engineer or his colleagues, technical standards and good practices, the state of the software, how it works, what it is best suited for, or even what would appear as the quickest, easiest or laziest option at the moment. Although there is a community of practice and exchanges between engineers, they usually work rather autonomously, with little external control on those choices.

However, the main source of local choices by the engineer is the codification process itself. What happens at this step is very close to what has been highlighted by quantification sociologists since Alain Desrosières (1993/2010) seminal work on the history of statistics. In its principle, codification or classification is the process that gives a label to an element such as an event, a document, an organism, etc., according to certain norms or to intuition. Desrosières has shown how difficult it is to consider classification as a mechanical procedure. By comparing Buffon and Linnaeus taxonomies, respectively top-down and bottom-up, he demonstrates that there is never a “right” codification. In his example of the attribution of a cause of death, he points out how physicians struggle to decide what cause should be written down, when there is usually an immediate cause, but also anterior reasons and multiple factors.

In text codification, similar issues arise. Instead of medical expertise, a sociological and linguistic one are required. Many moments of decision occur in the making and the attribution of codes. Codes must reveal relevant information for the analyst, but they also have to reflect the

actual content of the documents. In Proxem's context, they must refer to a customer's business concern such as "what can I do to improve my product/service ?", hence codes such as causes of disappointments, improvement suggestions… but such concepts must have a quantitatively significant presence in the data. As analyst actually go through the documents and read (at least part of) them, their code suggestions are as relevant as those who come at the request of Proxem's clients. As a result, Proxem code creation approach is both top-down and bottom-up. In the traditional methodology of pollsters, this step usually is completed autonomously by the analyst who works by following some conventions (each code should cover about 5% of the documents, the "other" code should represent no more than 2-3% of the data…) but also according to his expertise or point of view (Marc 2001). It is therefore quite opaque since the analyst is the only one accountable for both the creation of the codes and the codification of the documents. In Proxem's projects, the definition of the codes is the result of a negotiation between the analyst and the customer, by which bottom-up and top-down point of view can be combined.

The process through which a compromise is achieved is long and relies upon the idea, at least for the analyst, that there is no "ideal" coding plan towards which they are moving closer. Proxem's analysts for this step are mostly computational linguists, which means they have a background in both computer science and linguistics. As linguists, there are aware of the fact that words and phrases have no fixed meaning, and can therefore relate to several concepts and codes. They consider that their work relies deeply on intuition, common sense, and negotiation. They know that, although two analysts working on the same data would find common concepts and ideas, a significant part of it would be singular, or labelled differently. Over time, they develop their own "style" in crafting concepts and coding plans. In addition to the technical skills and the linguistic know-how of the computational linguists, there is a specific art of negotiation with the customer to come up with a good coding plan, that reflects both the customer's expectations and the actual data. This collaborative exercise of designing the coding plan was ultimately formalized into workshops and working documents

During the discussion with the customer, properly ontological questions may arise, leading to some kind of "ontological engineering". For example, a shoe seller wanted Proxem to detect some properties of their products such as "durability" and "abrasion resistance". When Proxem's computational linguists tried to find documents that matched with these concept, they wondered how one can express a concept without the other ; more precisely, if a shoe resists abrasion, doesn't it mean it is durable ? Conversations between analysts and with the

customers often deal with matters of what is a concept, what it resembles and what other concepts it includes, which are matters of ontology and mereology. Unlike formal ontologies such as the Linked Open Data project, Proxem's taxonomies handle concepts which are (re)defined for each project, implying that a word does not have a fixed, stable meaning because it depends on the context where it is used.

For those reasons, Proxem's computational linguists usually bring the bottom-up point of view for a project and consider that a coding plan is a neither necessary nor sufficient methodological construction, that could be different, although not completely arbitrary. They can be both nominalists, considering codes are labels put on things to organize them, and constructive realists who believe the coding plan is an acceptable, adequate view of the data. They are relativists, although not radical ones, for whom the coding plan does not pre-exist and does not have not be "found" or "remembered" like Platonic Ideas, but must be built in a fitting way.

Proxem's customers usually have a more realist point of view, and a top-down approach where they already know at least some of the codes that they want to be found and that make sense from a business angle : for them, the validity of a code comes from its ability to make sense for them and whether it will be possible to turn them into "actionable insights", that is knowledge relevant for business decision. As a consequence, the coding plan has a more confirmatory function, where codes act as measurements of issues such as customer satisfaction, process efficiency, brand reputation, etc. However this vision tends to evolves when they realize the coding plan must be a compromise between what they want to measure and what the documents actually contain and reflect. In the same manner, Proxem's computational linguists become pragmatists, in both the common and the Peircean sense : they try to please, or at least, find an agreement with the customer (common pragmatism where one tries to find "what works" rather than "what is right") and see as a criterion of validity (of the coding plan and the resulting analysis) the fact that the customer can take decisions on the basis of their result ; from this point of view, knowledge is assessed from its practical consequences and its ability to be a guide of conduct, as in Peircean pragmatism.

I have shown here that, in linguistics as in biology or medicine, a taxonomy can hardly be conceived from an idealist perspective, as an absolute, pre-existing the analysis and independent from the data. It is not either a purely empirical result deduced from the data, since choices have to be done and none of them are necessary. I will now describe how the coding plan and the data are connected, and how this connection is made.

In manual codification, the connection is assumed by the analyst. By reading each document, he determines what the document is about and if a fragment of it matches one or several codes ; he then assigns each relevant code to the document. The quality of codification depends on its judgment and on the consistency of his work. One of the benefit of automated codification is to improve consistency (and in the long run, speed). To this end, the automated approach as practiced by Proxem relies on the identification and writing of explicit rules of analysis, which produce “named entities”. Named entity recognition is a very common task in natural language processing. A very simple rule is the presence of a single word in the document : for instance the word “durability” in a document will activate the corresponding code. However, codes usually have a polarity (there convey a positive or negative opinion) that a single word can rarely grasp. Phrases as “disappointing durability” and “good durability” could activate corresponding codes, so there could be rules such as “durability after certain adjectives” which should be listed. However, in reality, people scarcely use phrases such as “disappointing durability” (so the rule would be useless) but they can say something like “durability is not good”, which reverses the polarity of a “good + durability” combination. Besides, when used alone, “durability” usually has a positive meaning. All in all, there are many ways to convey the same idea (including, in our case, without mentioning the word “durability”) and the work of Proxem’s computational linguists is to find those ways and to capture them with as few rules as possible.

Those rules are stored in a database which is common to all projects. This way, computational linguists do not have to write their rules from scratch for each project ; the theoretical justification for this approach is that, although each context and each project is different, there are similarities in meaning and in codes that can be used. For example, the way one expresses the durability of a shoe shares probably similarities with how the durability of clothes is described ; thus the rules written for a shoe seller may be re-used for a clothes seller. The pragmatic context of enunciation plays a key role in the transferability of rules. Similar contexts require similar rules, hence a customer complaining about his shoes resembles a lot a customer complaining about a piece of clothes. Although differences between projects and customers are well known, infinite differentiation would prevent any analysis from coming to an end, and a methodological individualism where each text is considered unique and incommensurable to another must be avoided. The underlying theory of knowledge that is at stake here is that, although it is desirable to reach perfect accuracy, an acceptable approximation is sufficient. The uncertainty lies henceforth on the strictness of this norm (that is, how good the approximation must be), which is not binary and may be adjusted according to customer

expectation, deadlines and budget. This way, the project's financial resources are as a matter of fact a good measure of accuracy, so that budgetary constraints rule epistemological strictness to some extent.

However, analyses are not purely custom-made and thus not entirely bound by material constraints. Computational linguists can draw on linguistic resources that were built independently from projects – and are adjusted to the current data afterwards. Those resources are a melting pot from several pre-existing sources, including the WordNet project, Wikipedia, geographical names databases, Linked Open Data. Those resources are usually established in a realist framework where those informational artefacts are regarded as a true representation of the organization of concepts they describe. It is very rare not to find at least ambitions of universality and belief in universals among ontology builders (Declerck & Charlet 2014). In this sense, it may appear that their theoretical framework is incompatible with the way automated codification is implemented by Proxem, and even more so with purely statistical approaches such as terminology extraction, a NLP technique to find statistically relevant terms (of one or more words) in a given corpus. The only way semantic web ontologies can work is on the realist assumption that the meaning of concepts is (mostly) universal, hence can be shared across the world.

On the contrary, statistical linguistics relies on a purely lexical approach to language (Pincemin 2012), without domain knowledge or any reference to the outside world ; they usually need only very low level language modelling such as what a word is (in European languages, a word is a series of characters between white spaces or punctuation marks[1], but in some languages neither is true). Named entity recognition has an intermediary level of formalism. Other techniques used by Proxem such as supervised classification, clustering, indexing, and later operations of more classical data mining all have different representations of text and language. All in all, the only way for those distinct representations of language to be theoretically reconciled, when corresponding sources are aggregated in the resources database, is through the epistemological opportunism of the Feyerabendian "anything goes". Resources are imported without their respective epistemologies and cast in the mould of the present theory of knowledge. Information is stored but meaning comes later, when resources are confronted to a new corpus and its coding plan ; knowledge is lost through storage, and restored when used, as usually in knowledge engineering (Bachimont 1996), for the coding plan.

Before we go further into the journey of data, it must be noted that those theories of knowledge are never explicitly claimed nor formalized by the engineers. Would they be questioned of this matter, they would

answer that they do not have a theory of knowledge, that they do not know what a theory of knowledge is, and would rather consider themselves as engineers, not scientists. The epistemological anarchism that I describe is not a explicit or voluntary stance, but something that can be deduced or interpreted, based on their attitude towards data and engineering.

After this clarification, we can note that the construction of meaning does not stop at the coding plan, and goes through more “theories of knowledge” as defined above. Mining operations performed on codified data no longer deal with language but with codes, or classes, which are meant to represent concepts, whose weight (based on the number of documents that match a code) needs to be trusted for the analysis to work (Porter 1995). Data mining uses those codes as a working representation of the initial phenomena ; the result of semantic analysis is the basis of data analysis. At this step, codes have become data and thus gain their own positivity, because the only way they make sense is if the analyst trusts them to do so. While computational linguists can be strong nominalists, analysts have to be realists in regards to the coding plan and the weight of codes. The statistical realism of Proxem analysts, is not, however, a naïve one : it is a probabilistic one, aware of measuring errors (where “measuring” actually is the coding process) and based on the idea that numbers are approximations that grow stronger in liability as volume increases. Based on the principles of exploratory data analysis (Tukey 1977), data mining on corpora ultimately relies on visualisation and storytelling (just as in history according to Ricoeur (1983)) to make sense and deliver knowledge in the pragmatic sense, that is assertions which can dictate action.

The conclusion of a data analysis can be expected to be a little less strict and literal to convey more perspective. However, in automated codification as practiced by Proxem’s computational linguists and analysts, perspective seems to be the rule. During the codification process as well as in further data mining, meaning and knowledge is never the purely mechanical result of a computation from presumed objective data. Many moments of choices, individual decisions and change of epistemic regime occur. Whatever the importance that the current doxa gives to algorithms in knowledge production, here they remain techniques that actualise possibilities, tools that give substance to models, theories of knowledge and personal judgment. The input data is not just a construction : the whole process is a process of data construction under several epistemic regimes :

- statistical sampling and opinion polling in social sciences govern the initial data ;
- text classification has a statistical and linguistic framework which

includes lexical, semantic and pragmatic levels, with flattened imports from knowledge engineering and artificial intelligence ;

- data analysis draws both on computational and statistical culture (data mining, exploratory data analysis, knowledge discovery in databases…) and the humanities to create a data-driven narrative.

However, epistemic regimes hardly govern all decisions, since many of them are also the combination of common sense, technical constraints and material conditions. They are rather rules to be compromised with facts, than norms. Coding plans as well as data narrative are the result of a social epistemology : the production and validation of knowledge is not the responsibility of a single individual but a social and collective one, shared across computational linguists, analysts, and end-users (namely, Proxem's customers), who ultimately bear most of the accountability in knowledge validation. Epistemic agents are successively or simultaneously realists, constructivists, pragmatists and relativists. They are continuously driven beyond their epistemic culture (Cetina 2000) : computational linguists grapple with philosophical questions on the meaning of words on a daily basis ; computer scientists implement linguistic and social sciences concept ; business analyst write linguistic resources and customers try to make sense of all this. Methodological strictness is no guarantee since different and not entirely commensurable (hence partially contradicting) methods are in use.

However, the resulting narrative is not a mere rhetorical construction for all that. Although the whole process contradicts the idea of absolute truths, and has only an instrumental tool of mechanical reason, it actually produces sufficient approximations of pragmatic knowledge. Through educated guesses and multiples methodological loans, epistemic agents manage to make sense of text and data. To do so, they rely on a specific know-how that comes from experience and dexterity ; each of them has their own style, their own habits. Text codification is more craft than science, more skill than software. Based on the novelty of big data exploitation as a field and an industry in general, I tend to believe that as for now, data sciences in general would be better called "data crafts", or more precisely, that most of the work currently done with data falls under this category. Not only does it take the majority of time and energy of the so-called data "scientist", but it does not seem that this part of the process will disappear in the foreseeable future. It does not seem either that this disappearance would be advisable, since this is also where most of the understanding and sensemaking of the data happens, through the informed choices of the engineers and analysts. Data manipulation would then rather be studied and explained through the analysis of human processes and agent behaviours, than through the current focus on software and

algorithms which tends, as it seems for now, to hide individual decisions, arbitrariness and complexity.

Bibliographie

Bachimont, B. (1996). *Herméneutique matérielle et artéfacture, Des machines qui pensent aux machines qui donnent à penser*. Thèse de doctorat en épistémologie de l'Ecole Polytechnique.

Cetina, K. K.(2000). *Epistemic Cultures : How the Sciences Make Knowledge*. Harvard University Press.

Cailliau, F., & Poudat, C. (2008). *Caractérisation lexicale des contributions clients agents dans un corpus de conversations téléphoniques retranscrites*. JADT 2008, 267–275.

Desrosières, A. (2010). *La politique des grands nombres. Histoire de la raison statistique*. La Découverte.

Declerck, G., & Charlet, J. (2014). « Pourquoi notre sémantique naïve n'est pas formalisable et pourquoi c'est (presque) sans conséquence sur l'ingénierie ontologique. » *Intellectica*, *1*(61), 1–46.
Frippiat, D., & Marquis, N. (2010). « Les enquêtes par Internet en sciences sociales : un état des lieux. » *Population*, *65*(2), 309–338.

Grossetti, M., & Boë, L.-J. (2008). « Sciences humaines et recherche instrumentale : qui instrumente qui ? » *Revue d'Anthropologie des connaissances*, 2, 1(1), 97.

Marc, X. (2001). « Les modalités de recueil des réponses libres en institut de sondage : le rôle de l'enquêteur, les consignes et les procédures de contrôle, les perspectives d'amélioration. » *Journal de la société française de statistique*.

Pincemin, B. (2012). « Sémantique interprétative et textométrie. » *Texto ! Textes et Cultures*, XVII, 1–21.

Porter, T. M. (1995). *Trust in Numbers*. Princeton University Press.

Ricoeur, P. (1983). *Temps et récit 1*. Seuil.

Tukey, J. W.(1977). *Exploratory Data Analysis*. Addison-Wesley.

---

1 With some exceptions that we will not list here. For example, acronyms where letters are separated by periods, such as "N.A.T.O.", are considered words by good tokenizers. Punctuation is not always the limit of a word.

+

Schmitt, Eglantine. « Explorer, visualiser, décider : un paradigme méthodologique pour la production de connaissances à partir des big data. (Doctorat) », 14 janvier 2019, Cahiers COSTECH numéro 2.
http://www.costech.utc.fr/CahiersCOSTECH/spip.php?article97