Article

Improving Access to Justice with Legal Chatbots

Marc Queudot [1,2], Éric Charton [2] and Marie-Jean Meurs [3]* 0000-0001-8196-2153

[1] University of Quebec in Montreal UQAM; queudot.marc@courrier.uqam.ca
[2] National Bank of Canada; eric.charton@bnc.ca
[3] University of Quebec in Montreal UQAM; meurs.marie-jean@uqam.ca
* Correspondence: meurs.marie-jean@uqam.ca

Version July 16, 2020 submitted to Stats

Abstract: On average, one in three Canadians will be affected by a legal problem over a three-year period. Unfortunately, whether it is legal representation or legal advice, the very high cost of these services excludes disadvantaged and most vulnerable people, forcing them to represent themselves. For these people, accessing legal information is therefore critical. In this work, we attempt to tackle this problem by embedding legal data in a conversational interface. We introduce two dialog systems (chatbots) created to provide legal information. The first one, based on data from the Government of Canada, deals with immigration issues, while the second one informs bank employees about legal issues related to their job tasks. Both chatbots rely on various representations and classification algorithms, from mature techniques to novel advances in the field. The chatbot dedicated to immigration issues is shared with the research community as an open resource project.

Keywords: Chatbot; Information Retrieval; Natural Language Processing; Question Answering System; Dialog System

1. Introduction

The non-representation of a high proportion of litigants is one of the most striking factors contributing to problems of access to justice. Laniel *et al.* [1] addresses this problem and highlights the unintended side of this situation for many litigants, while judges treat "self-representation" as a choice, if not a privilege. To reflect the fact that this situation is often not a choice, the authors prefer to refer to it as "non-representation". Schneider [2] describes the situation in the United States, where the Supreme Court has ruled that requests submitted by a Non-Represented Litigants (NRLs) (*pro se complaints*) should be subject to a lower level of requirement than motions brought by counsel. According to the author, the criteria used by the courts to determine the validity of claims further disadvantages NRLs. The onus is on the plaintiff to show that the complaint is credible, which is difficult for those typically under-resourced. This gap is even greater where preliminary research is required, or where the opposition holds all the key elements of the case. According to Schneider [2], the constraints of building a case and studying pleadings jeopardize the right to represent oneself in court and even the constitutional right of citizens to an opportunity to be heard.

Beyond the high costs of consultation and representation that hinder the representation of litigants, other obstacles make access to justice difficult, such as limited access to legal information, but also geographical constraints, etc. Our contribution in the form of an immigration chatbot could facilitate access to information for litigants in this area. The work presented in this article was carried out as part of the LegalIA project[1]. This project brings together researchers from Université du Québec à

[1] https://legalia.uqam.ca/

Montréal (UQAM), Concordia University and Université de Montréal around the issue of the ethical and responsible development of the law. To present the project, we produced a video[2] illustrating the need for such initiatives. The first part of the proposal is to create a chatbot to assist the actors in the legal field: lawyers, judges and litigants. This tool will then constitute a case study for the sociological and ethical analyses in the second phase of the project. In practice, the contribution presented here approaches the part intended for individuals, as opposed to lawyers and judges, as described in the video. There are many other initiatives that address different facets of the access to justice problem. Some associations offer free legal aid to the most disadvantaged. In Montreal, for example, clinic *Droits Devant* accompanies homeless people, and helps them protect their rights, particularly during criminal proceedings. In several universities, students have access to free legal advice (e.g., at Université de Montréal and UQAM), provided by law school students under professional supervision. Since March 2018, the website *La boussole juridique*[3] lists organizations which offer legal services for free or at a lower cost.

In Quebec, *Éducaloi*[4] brings together experts from many legal fields and produces concise guides to inform citizens about their rights. These guides facilitate access to legal information by popularizing it and organizing it into more user-friendly elements which better correspond to what people are looking for than would a simple popularization of each piece of legislation. It is in this context, and particularly considering the problems of access to legal information, that our work fits in. The global context of our research is that of access to justice and how to improve its current state using AI-powered technology, but in this work, we focus on access to information. In practice, we develop Information Retrieval (IR)-based chatbots in two different settings. The first is targeted at the general public in need of immigration information. The second is an internal resource for the employees of National Bank of Canada (NBC) who have questions regarding the legality of different aspects of their jobs.

First, we introduce in Section 2 recent developments in chatbot technology in the areas we are interested in. Next, we explain in Section 3 the methodology we followed in this work. We describe in Section 3.2 the data used, as well as the problems we encountered, especially in collecting them. Section 4 reports on the experiments as well as their analysis. Finally, the last Section summarizes the work carried out and presents future steps.

2. State of the Art

In this Section, we introduce important work and recent best approaches in the areas we are interested in. To properly understand the approaches used by the proposed systems and their limitations, it is necessary to know the metrics used to evaluate them. We therefore begin by defining them. Next, we introduce predictive justice approaches, and we end with dialogue systems. For more details on the presented techniques, we recommend these two reference manuals: [3] for Natural Language Processing (NLP) and [4] for IR.

2.1. Evaluation Metrics

In IR and classification, the following metrics are often used for binary classification. For each metric, one can calculate the results by class and aggregate them afterwards if the problem has more than two classes. Instances of the positive class, correctly classified, are called True Positives (TP). True Negatives (TN), False Positives (FP) and False Negatives (FN) are defined in the same way.
The **precision** is the proportion of truly positive instances among those identified as positive by the system: precision $= \frac{TP}{TP+FP}$.

2 https://legalia.uqam.ca/legalia/
3 http://boussolejuridique.ca/
4 https://www.educaloi.qc.ca/

Table 1. Confusion matrix for a three-class classification problem

		predicted A	predicted B	predicted C
true	A	3	1	1
true	B	4	5	1
true	C	0	1	6

The **recall** is the proportion of truly positive instances that have been correctly classified by the system: recall $= \frac{TP}{TP+FN}$.

Figure 1[5] illustrates which part of the dataset precision and recall refer to.

The **accuracy** is the proportion of instances, both positive and negative, that were classified correctly: accuracy $= \frac{TP+TN}{TP+FN}$.
Accuracy can be used if the classes are well balanced, but will give biased results towards the majority class otherwise. For example, with a 90/10 distribution of instances between the positive class and the negative class, the rule "always assigns to the positive class" would get 90% accuracy.

The **F-measure $\mathbf{F_1}$** is the harmonic mean of precision and recall: $F_1 = 2\frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$.
In general, this metric is the best choice if the classes are not well balanced.

Each of the aforementioned metrics is defined in the case of a binary classification, and must be aggregated to evaluate a multi-class classification system. Then, three different averages are used: the *micro, macro and weighted* averages, which are defined and illustrated hereafter.
For example, consider a classification task with 3 classes A, B and C where there can be only one class for each example, the results provided by the classification system being reported in the confusion matrix from Table 1.
In the case of the **micro-average**, one evaluates all the classes together. For the **micro-precision (microP)**, one counts the total of TP and FP using the confusion matrix:
$TP = TP_A + TP_B + TP_C = 3+5+6 = 14$, $FP = FP_A + FP_B + FP_C = 4+2+2 = 8$, then one calculates the micro-precision $= \frac{\text{TP}}{\text{TP+FP}} = \frac{14}{14+8} \approx 0.64$.

For the **macro-precision (macroP)**, the precision for each class is calculated (which is noted p_A to p_C), then the average is calculated. Using the same example, one has $p_A = \frac{\text{TP}_A}{\text{TP}_A+\text{FP}_A} = \frac{3}{7}$, same for p_B and p_C. Macro-precision is therefore: $(p_A + p_B + p_C)/3 == (\frac{3}{7} + \frac{5}{7} + \frac{3}{4})/3 \approx 0.63$

The weighted average assigns a weight to each class equal to the number of instances associated with it. In our case of 22 instances, they are respectively 5 instances of A, 10 of B and 7 of C. The **weighted precision (wP)** is therefore $(p_A \times 5 + p_B \times 10 + p_C \times 7)/3 = (\frac{3\times5}{7} + \frac{5\times10}{7} + \frac{3\times7}{4})/22 \approx 0.66$.

Notation: In all this document, a variable in bold, for instance **v**, denotes a vector.

The **cosine similarity** between two vectors allows to compare these two vectors with each other, being equal to their scalar product divided by the product of their Euclidean norms. With two vectors $\mathbf{v_1}$, $\mathbf{v_2}$ and θ the angle between $\mathbf{v_1}$ and $\mathbf{v_2}$:

$$\text{sim}(\mathbf{v_1}, \mathbf{v_2}) = \cos(\theta) = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{|\mathbf{v_1}| \times |\mathbf{v_2}|} \quad (1)$$

[5] Contribution by a Wikipedia author https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg, visited January 2, 2020, adjusted for landscape format.

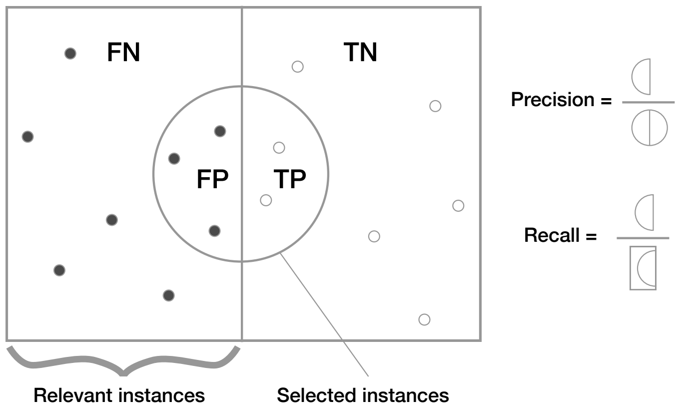


Figure 1. Graphical interpretation of accuracy and recall

Table 2. Example of BoW representations with a small corpus

	a	all	am	apply	can	for	foreign	get	how	I	on	online	student	to	visa	while	work
A	1	1	0	1	1	1	1	0	0	0	0	1	1	0	1	0	0
B	1	0	0	0	0	0	0	1	1	0	1	0	1	1	1	0	0
C	1	0	1	0	1	0	0	0	1	1	0	0	1	0	0	1	1

2.2. Representation Learning and Text Classification

One of the main sources of performance improvement in NLP is the way the text is represented so that it can be processed by the different classification algorithms. To understand the value of current techniques, it is important to put them in context. This Section describes the main methods that have been used over time. We will see later that some of them, although developed several decades ago, still make it possible to design competitive models in various contexts.

2.2.1. Bag-of-Words and Variants

One of the most naive ways to represent a textual document is to use the unordered set of words that make it up. This technique is called Bag-of-Words (BoW). The representation of a particular document depends on a dictionary of terms, which is usually built on the whole corpus of available documents. For each document, a vector is built, with coordinate values of 0 or 1 depending on the presence or absence of each word in the dictionary. The binary variables can also be replaced by the number of occurrences of each word. In both cases, this leads to the production of sparse vectors, because their coordinates consist mainly of zeros. A toy example is given using the following very short documents:

A: "`All foreign students can apply for a visa online.`",
B: "`How to get on a student visa?`",
C: "`How can I work while I am a student?`".

Table 2 shows the BoW for these sentences, representing the only three documents used to build the dictionary. In this example, the sentences have been normalized by keeping only the lowercased lemmas, so that for instance, apply and applicant are grouped under the same term "appl*". With a reasonable number of documents not dealing with exactly the same topics, the number of terms in the dictionary grows rapidly, while the number of ones in each vector hardly changes. For example, if the dictionary has 10 000 words, there are only 15-20 non-zero values in each vector of size 10 000, for documents the size of an average sentence. The large dimensionality of these representations implies a high computational cost for many algorithms, but the fact that the vectors are sparse allows for the development of specific techniques to reduce this cost. These techniques will be presented in Section 2.2.2. These representations are useful for any task in NLP, as input to classification algorithms as described later, or to directly compare documents with each other, using for example the cosine similarity defined in Section 2.1.

A major disadvantage of these representations is that they give equal weight to all words. They do not take into account either the length of the document or the fact that certain words are present in the vast majority of documents. Indeed, a very long document has a high probability of containing any word, but this word has a lower impact on the subject of the document than in a shorter document. Similarly, very frequent words (pronouns, for example) provide little information about the subject of a document that uses them. The statistic called Term Frequency-Inverse Document Frequency (TF-IDF) calculates a score for each word that takes into account the two points we have just described (document length and word frequency). The frequency of the term in the document (TF) weights the number of occurrences of this term by the length of the document. This component is defined in Equation 2, where $f_{t,d}$ is the number of occurrences of the term t in the document d.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2}$$

The IDF component is a heuristic introduced by Karen Spärck Jones [5] on the intuition that a term (a word) appearing in many documents does not allow to properly discriminate between them. This score for a term t_i, which appears in n_i documents among N documents, is calculated as defined in Equation 3.

$$IDF(t_i) = \log \frac{N}{n_i} \tag{3}$$

Each of the three sentences:
I: "`I received my work visa.`",
J: "`I need a student visa`",
K: "`I received a letter.`",
would have BoW representations equally different from the other two. Indeed, the word `visa` is common to I and J, the word `a` is common to J and K, the word `received` is common to I and K and the word `I` is shared by all of them. Since `a` and `received` are very common words compared to `visa`, the score associated with the latter would be higher than for the other two words with the TF-IDF representation, so I and J would be the closest pair among the three possible. Many authors have sought to give a strong theoretical justification for the intuition of Spärck Jones [5], often based on the information theory of Shannon and Weaver [6]. According to Robertson [7], one can find many limitations to these a posteriori interpretations. Nevertheless, Robertson [7] acknowledges the impact of this statistic on the field of NLP, being the key resource of almost every search engine, among other applications.

2.2.2. Word Embeddings

Vector representations produced by BoW have the advantage of being simple to understand and quite efficient. However, they also have limitations. Here, we reuse the example sentences A, B and C from Section 2.2.1. If we replace the word `visa` in document B by its synonym `permit`, the resulting document B' : "How to get on a student permit?" has no more words in common with sentence A (with the exception of `a`, which is shared among all documents).

Using cosine similarity on their BoW or TF-IDF representations, these documents would be considered entirely different, while the similarity between A and C, and between B and C, is non-zero. However, for most classification tasks, obtaining similar representations for semantically close words is important. *Word embeddings* are among the approaches where vector representations of words are

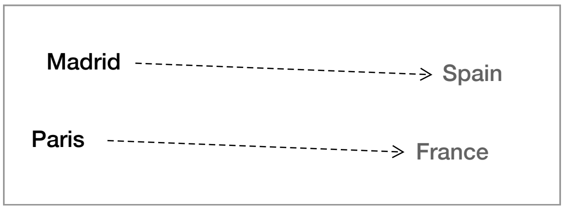


Figure 2. Word embeddings of city and country names

learned from their context, rather than explicitly defined as previously described. The distributional hypothesis introduced by Harris [8] suggests that words that are semantically close tend to be used in the same context. While latent semantic analysis represents the text by identifying it with the topics to which it relates [9], word embeddings use words as context. Mikolov *et al.* [10] introduces two techniques for learning their representations from their surroundings.

Even though these two techniques are not the first to allow the learning of distributed representations of words, the light structure of their model allows for the first time for training on very large datasets with limited resources. According to Mikolov *et al.* [11], representations for 100 billion of words can be learned in one day calculation on a single machine. Mikolov *et al.* [11] also show that these representations contain not only semantic information about words, but also syntactic features. What is surprising is that it is possible, in this representation space, to perform linear translations (displacements) that correspond to these characteristics. Figure 2 illustrates that the embeddings of the cities are close to each other, and those of the countries are too, but more importantly, the relative position of "France" compared to "Paris" is similar to that of "Spain" compared to "Madrid".

Since the words `visa` and `permit` are semantically close and should often appear in the same context, their embeddings are very similar. In our example, the following relationship holds true: $\mathrm{sim}_{emb}(A, B') \sim \mathrm{sim}_{emb}(A, B)$. If we used word embeddings and simply averaged or summed all vectors to create a sentence vector, the similarity between any two sentences among A, B, C would be similar since they all have exactly two words in common with the others. Using a weighting system [12] or an embedding technique designed for sentences [13,14] would allow the similarity between A and B to be the largest, which makes the most sense since they are both about student visas.

2.3. Language Models and Recurrent Networks

While word embeddings are trained with the explicit goal of retaining semantic information in their representations, *language models* learn it implicitly. Indeed, the objective of language models is to capture the probability distributions of word sequences. Since the nature of language is not well understood and the universe of possible word sequences is extremely large, language models use certain simplifications to approximate the probability of appearance of a sentence. For example, n-gram models maintain a count of occurrences of word sequences of n length or less. To calculate the probability of appearance of a sequence of length $m > n$, one simply multiplies the probability of each of the n word sequences it contains. Once trained, these models can be used in concrete applications to represent words as one would do using word embeddings.

The first language patterns were learned using non-recurrent neural networks [15,16]. To handle contexts of variable size, one now uses recurrent *encoder-decoder* architectures instead [17]. The first of the two networks, the *encoder*, is a Recurrent Neural Network (RNN) which reads the words of the sentence to be translated one-by-one, and predicts the next word at each step. Once the sentence is complete, the hidden state contains a summary of the sentence. That summary, also called the *context* is then fed to the second network, the *decoder*. This model then generates an output sequence based on that context, but also the previous outputs as it goes on.

These two steps of the recurrent encoder-decoder are illustrated by Figure 3. For each symbol being read, the encoder updates its hidden state using the previous state as well as the current symbol.

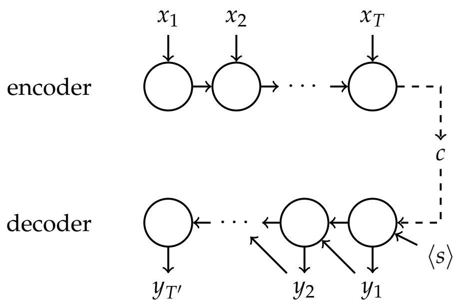


Figure 3. Encoder-decoder sequence to sequence model

Once the last symbol has been read, the network has the context in its memory. Each symbol is predicted by the decoder using the context, the previously predicted symbol and the current hidden state. The latter is itself derived from the hidden state and the symbol of the previous time step. In our example, machine translation is the task used to jointly train the encoder and decoder models. While this model can be used as-is, the pre-trained encoder can then be used independently in other tasks with smaller corpora to represent documents.

The recurrent encoder-decoder is based on a variant of RNN, which uses the recurrence mechanism to model the order of words (or symbols) in the text, while *Transformers* are a new type of neural networks which use the attention mechanism instead of recurrence. This mechanism introduced by Bahdanau *et al.* [18], originally for automatic text translation, has become the source of performance improvements for many reference tasks in NLP. The general idea is to predict, at each timestep (token) in a sequence, the importance of each of the elements passed as input to the network. This is referred to as the *attention* that the network attaches to particular elements of the input sequence. In the case of translation, one part of the network learns the translation part *per se*, and the other part learns to align the words or groups of words. Transformers [19] use Convolutional Neural Networks (CNNs), combined with the attention mechanism. This technique completely eliminates the sequential dependency of recurrent networks, and thus allows to fully parallelize aining. Although their training has a much higher overall computational cost than their equivalent in recurrent networks for example, this advantage makes them faster to execute, provided one has access to large parallel computing resources. As with word embeddings (see Section 2.2.2), what allowed the popularization of transformers is the public availability of pre-trained networks, which can be used as-is to represent sequences of text, or fine-tuned to specific tasks in a reasonable amount of time to achieve even better performance. Devlin *et al.* [19] introduced a transformer-based architecture called Bidirectionnal Encoder Representations from Transformers (BERT), whose pre-trained models are now very popular.

2.4. Chatbots

Chatbots could support the improvement of access to legal information. In this Section, we describe some important contributions in this area. Chatbot systems can be divided into three different categories, Question Answering Systems (QASs), "social" chatbots and focused dialog systems. The main characteristic of a QAS is the use of data of various types, such as web pages or knowledge graphs, to provide direct answers to user questions. Social chatbots are rather generalist systems, while focused dialog systems are usually dedicated to a specific tasks.

2.4.1. Question-Answering Systems

Search engines or other search systems return documents in response to user requests. These queries can consist of a combination of keywords, but can also be in natural language (without any particular constraints, notably in terms of vocabulary). As an example, a user seeking information on work authorizations could formulate the following queries:

- Student visa, work authorization.

Table 3. Database table of a knowledge-based QAS

nationality	type	requirements
american	work	You need a job offer.
american	study	You only need a document mentioning the equivalent degree.
american	PVT	This visa is not offered to American citizens.
french	work	You need a job offer and a report from a competitivity analysis.
french	study	You need to show an acceptation letter and have $10 000.
french	PVT	You need to register to the random drawing.

- I'm on a student visa. Can I work?

While a search engine would return essentially the same documents to both queries, a QAS could use the information from the natural language query to answer it precisely. Rather than having to read the complete documents describing student visa rules, the user of the QAS could receive an answer such as "As a general rule, students can work 20 hours per week during the semester, [...]". According to Jurafsky and Martin [20], QASs are of two types. The first type associates queries with logical representations to query structured knowledge bases. In the second, relevant documents are first searched for using IR techniques, and then text understanding algorithms are used to extract relevant Sections.

In the case of a QAS whose role is to provide information to immigrants, a simple knowledge base could be a table associating nationality of origin and visa type to a list of pre-requisites. We have the sets (incomplete but for the purposes of the example) $type = \{work, study, PVT\}$ and $nationality = \{american, french\}$. The corresponding database table is shown in Table 3. Figure 4 presents a flowchart of the interactions with this QAS to answer the question "What are the pre-requisites to get a visa?". The interactions with the user must inform the two strangers (nationality and type of visa) before they can be given an answer using the database Table 3. Since the system follows very simple rules, it is entirely predictable and, assuming the information is correct and complete, the correct answer will be provided to the user in all cases.

It would be possible to apply this paradigm in a second step to systems such as the ones we are developing, to further facilitate access to information. However, this paradigm is not flawless since it forces interactions to remain within a very rigid framework. The effort required to build the knowledge base is also much greater than that required to bring together an unstructured corpus of data. Work exists to extract structured information from relatively homogeneous textual data, notably Auer *et al.* [21] created an ontology from the set of Wikipedia pages and the relationships between them (the hypertext links). These approaches allow access to a large relational knowledge base, at the cost of a decrease in data quality compared to manual work. Despite this limitation, this type of tool is an interesting source of improvement for further work. In particular, we have begun its integration into the NLP tools of the NBC to increase their coverage thanks to the automatic extraction of synonyms from this database.

A IR-based QAS could be designed using a corpus that includes the documents below:

- d_a:"If you are an American citizen, you need a job offer to apply to a work visa. To get a student visa, you only need to prove your student status. There is no *working holiday* visa option.
- d_f:"If you are French, you will need [...]"

To answer the following queries r_1 and r_2, the first step consists of identifying the right document.

- r_1: "What are the prerequisites for a work permit for U.S. citizens?"
- r_2: "What are the prerequisites for Americans to get a work permit?"

This step can be performed using one of the methods presented in Section 2.1. A simple cosine similarity on BoW vectors would work with the r_1 query, which has vocabulary in common with d_a, but not with the r_2 query, which uses slightly different words. Trained embeddings would match these both queries with d_a.

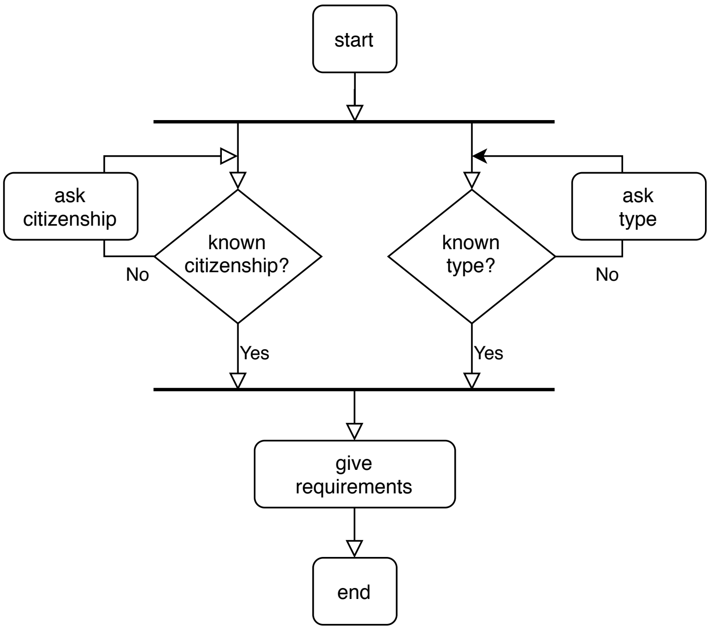


Figure 4. Flowchart of the "knowing visa requirements" part of a immigration chatbot relying on a structured knowledge base

Various techniques can then be used to extract the relevant part of the document to answer the question. One usually begins by applying a grammatical analysis of the query to determine the type of question ("Who", "What" "When", etc.). It is then possible to use morpho-syntactic labeling tools to filter the results and make a cosine similarity comparison for example to select the most relevant sentence(s).

2.4.2. General Chatbots

The design of chatbot systems not restricted to a specific task is difficult, and the evaluation of such systems is complex. The characterization of what constitutes a good conversation has inspired many authors but still remains an open research question [22]. The first generalist chatbot, ELIZA [23], was created to mimic a conversation with a psychotherapist. The algorithm simply consisted of associating certain keywords with predefined phrases. Thanks to Weizenbaum *et al.*'s choice of "persona", the system received very positive feedback from users, without taking great risks since it answered mainly with new questions.

The ALICE [24] system was the first to handle natural language entries. The Artificial Intelligence Markup Language was developed within this framework, to design a system of conversational rules. Many chatbots created using this language are listed in Satu *et al.* [25].

As with other areas of NLP, learning-based dialogue systems are currently the most common approach to conversational systems. Vinyals and Le [26] use a sequence-to-sequence neural network architecture (*seq2seq*, introduced by Sutskever *et al.* [27]) to train a chatbot from end-to-end without any prior knowledge. The model simply learns how to predict the most probable answer (character by character) to a question from the training corpus. The simplicity of this model has several advantages. The only data needed are discussions: no special annotation is required, so many corpora are usable. Moreover, since this method does not require any domain-specific knowledge, it is very easily applicable in a variety of contexts.

However, this approach has limitations, which the authors acknowledge. First, the objective of predicting the most likely next step in a conversation is not a very good indicator of the purpose of a conversation. Indeed, the goal of a conversation is often a longer-term objective such as sharing a certain piece of information or completing a specific task. Second, the model tends to favour low-risk responses, which often amount to very short and uninteresting answers. Finally, the model does not contain any mechanism ensuring the answers are consistent with each other. This last point is, according to the authors, one of the points that prevent their model from passing the Turing test.

Qiu *et al.* [28] explain that IR-based methods often fail to deal with long, precise questions. They also point out that text generation-based models may generate inconsistent or meaningless answers. Motivated by the limitations of both of these approaches, they developed a hybrid chatbot, which combines IR and text generation in a commercial setting, to provide customer service and shopping recommendation. This approach leads to three models: a IR model, a text generation model, and a third one selects a answer to provide a user with according to the confidence threshold of the first algorithm. When evaluated manually, the hybrid model outperforms the simple model with 60% of text accuracy versus 40%.

The Turing test, introduced by Turing [29] under the name of "Imitation Game", involves a machine trying to impersonate a human through a written conversation with an examiner. This test was Turing's precise way of answering the question "Can machines think?" through observation. According to Radziwill and Benton [30], this objective has guided the development of chatbots since ELIZA [23]. Ramos [31] as well as Radziwill and Benton [30] suggest that this ability to mimic human behaviour is not necessarily a desirable quality however, and argue that even human empathy towards these systems would not suffer from a lack of it.

Deriu *et al.* [32] list other competitions and metrics, which attempt to evaluate the quality of a chatbot system, but no evaluation method stands out as a de-facto standard. Metrics such as BLUE [33] and RED [34] measure the overlap between the chatbot dialogue and pre-set phrases. Some approaches, such as Lowe *et al.* [35], use a RNN to try to predict how judges would rate sentences or the entire conversation. This approach requires extensive manual annotation, but its results correlate quite closely with user judgments. Despite many promising leads, the problem is still open since it seems difficult to identify the quality factors of a chatbot in general.

2.4.3. Task-specific Chatbots

For task-specific chatbots, whether informational or transactional, the application domain is known in advance. In the later case, the role of the chatbot is to automate the exchanges in order to carry out a transaction: the cancellation of a subscription, or the sending of a bank transfer for example. This has allowed the development of finite state systems [36] and rules-based systems to govern transitions between these states. The state of a system is composed of variables and their values, which describe all the elements of the environment and chatbot configuration needed to guide the chatbot decisions. In our very simple example, illustrated in Figure 4, a state of the system could be described as follows:

```
started: TRUE; citizenship: UNKNOWN, visa_type: STUDENT; requirements_given: FALSE
```

When users provide their citizenship, the property `citizenship` is now set (to `FRENCH` for instance) and the updated properties and their values now constitutes the new state.

These successful techniques, coupled with statistical approaches [37,38], are still in use today because they are capable of supporting the design of simple and robust systems. The system developed by Bobrow *et al.* [36] is still the basis of many flight reservation systems today [39].

However, these techniques have the disadvantage of not being flexible. Indeed, transitions between states are normally questions (or patterns of questions, as with ALICE). If several questions should lead to the same action, the rule must be duplicated. Other information may also be taken into account when choosing a state transition (e.g. the geographical position of the user or the number of days before the next visa lottery, where applicable). In general, with n binary variables, the size of the tree of possible states will be 2^n.

For this reason, it is often difficult keep track of states and their transitions. The concept of "intent" [40] was developed to help to remove the coupling from a specific user utterance with a state transition. The intent designates the desired outcome of the interaction. The two sentences "I want to know how to get a student visa." and "Requirements for foreigners to study in Canada" both convey a similar information need, which could be an intent called get_information_student_visa. The Rasa framework[6] uses intents and handles the state transitions in the following way. An intent classification model is responsible for assigning the correct intent to each user interaction, and then a second model uses the context of the conversation and that intent to identify the best response to that interaction. Assuming that there is no history with relevant additional information, a chatbot could answer with an action such as give_generic_student_visa_info. If additional information is known, the action could be personalized, for example, by returning information specific to French immigrants.

As we will explain in Sections 4.1 and 4.2, the systems we developed do not need to track any state in their initial versions, but could benefit from state tracking in future versions. While the intent classification task could be approached with many techniques, the StarSpace [13] algorithm is one of the most efficient supporting this task. StarSpace is an algorithm for learning embeddings for entities of different types in the same space, in a supervised manner. For intention classification, the entities are therefore the documents (user-generated text and Frequently Asked Questions (FAQ)), as well as the intent labels. The algorithm consists of matching positive document-intent pairs (those where the intent matches the document) and limiting the proximity of negative pairs. In practice, entities are represented by their features, and the representation of these features that is updated by minimizing the loss function by stochastic gradient descent. To optimize the training time, the authors use negative instance sampling [10] and a margin parameter in the loss function to avoid focusing on near-perfect instances.

The next Section will present the methodology we followed to develop two chatbots by building on these techniques.

3. Methodology

3.1. Experiment tracking and reproducibility

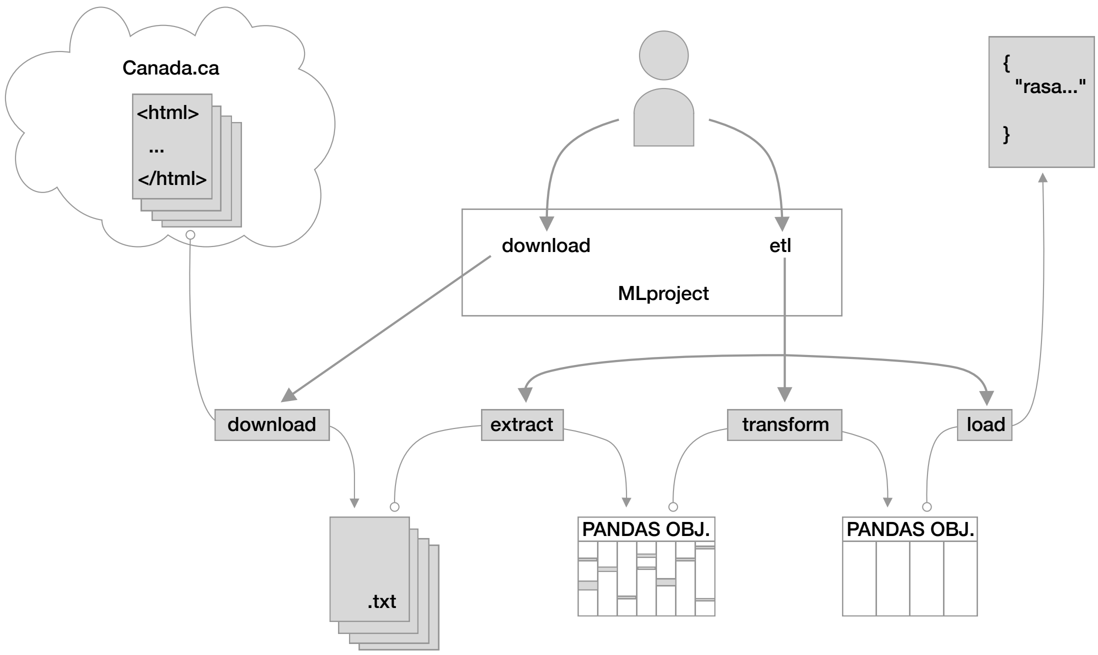
In recent years, the phenomenon of the "reproducibility crisis" in science has been much discussed. Baker [41] reports the results of a survey in which more than 1500 researchers are questioned about their experience in reproducing research work. More than 70% of the participants have already failed to replicate other researchers' experiments, and more than 50% have failed to replicate their own experiments. This study covers various fields of science but computer science is not spared. Gundersen and Kjensmo [42] report on a similar observation: they review 400 papers from top Machine Learning (ML) conferences[7], and find that most of these experiments are non-reproducible due to lack of documentation. Gundersen and Kjensmo [42] suggest that the format of scientific papers does not lend itself to the transmission of all the precise information needed to reproduce the experiments, as opposed to code. Sharing the code and model parameters would therefore allow an additional level of reproducibility when compared to simply describing the experiment in a scientific paper. Code version management is enabled by tools such as subversion or git. These tools intervene in several steps before publishing models and articles to describe experiments. The design of a learning-based system contains many interacting elements. Earlier, we discussed this topic to support the recommendation to share the source code of experiments, but before that, it often takes dozens of model training attempts. Managing these experiments is a complex but necessary task, for which there is no widely accepted solution such as code versioning systems.

We use the MLflow framework [43] for two elements of the model development process. First, for experience tracking, we keep track of each execution of the scripts for data collection, data preparation,

---

[6] https://rasa.com
[7] IJCAI and AAAI, ERA ratings through http://www.conferenceranks.com/.

Figure 5. Data collection and cleaning for the immigration chatbot, and developer interaction



and model training and evaluation. This tool allows us to go back later on to an experimental result, to find exactly its context in order to analyze and describe it, and to reproduce it, or to make new experiments from the same experimental set-up. We have also added a level of abstraction on top of each of the scripts we use (a MLflow *project*) to be able to call them simply with default values. These scripts allow a better repeatability of our experiments since only a few commands are needed to train the models we describe.

For the immigration chatbot for example, Figure 5 describes the interaction of a user with the data collection and preparation project. As can be seen, the user make use of only the two entry points exposed in the *MLproject* file. The first entry point triggers the download of the document collection, and the second one starts its processing. This data processing is decomposed according to the design pattern "extract, transform and load". The three steps are respectively data recovery from the source, data cleaning and formatting, and then export to the final format usable by the RASA framework.

3.2. Data

3.2.1. Immigration Canada FAQ

In an attempt to use chatbots to provide an easy way to access legal information, we set out to look for a suitable dataset. The ideal corpus for such a task would contain a large number of diverse conversations about a fair number of different topics. Conversations datasets are already hard to find to develop general-purpose chatbots though, and this problem is even more pronounced for law-related corpora. Indeed, many conversations would contain potentially sensitive information, which would make those datasets impossible to release. In cases were lawyers are involved, the attorney-client privilege protects the conversations even more strictly, so accessing this kind of data is not an option.

One source of legal information that is not typically considered as a conversational dataset is the explanations given by the government on administrative procedures. We have collected 1088 webpages in English from the Government of Canada's Immigration and Citizenship Help Desk[8]. This dataset presents the advantage of being public and exempt of any privacy concerns.

Together these pages constitute an immigration FAQ. Each page begins with a question associated with a category (e.g. "Studying") and is followed by an answer consisting of one or more paragraphs of text. The page also contains a list of questions that other users have found useful. Finally, two lists of keywords are also present in the page but invisible to the user. After verification, we have chosen not to extract these lists since it does not change from one page to another.

Once the documents are retrieved, we use the HTML structure to extract the information from each page. The data collection and preparation code is made available[9] to facilitate the reproduction of the results.

Our dataset consists of 1088 classes with only one (1) example per class. For learning-based models, these examples will constitute our training dataset. In order to guide the optimization phase of our system and perform a preliminary evaluation, we have created paraphrases of 88 examples of the training dataset. With this set of validation data, we can validate our hypotheses and quickly iterate on the system design. This dataset would benefit from a larger volume by increasing the number of annotated examples to evaluate the system performance more precisely. However, for the proof-of-concept we present here, this small corpus is sufficient.

3.2.2. Internal Legal QAS

Our second chatbot is developed within the NBC to answer employees' legal questions. As for the immigration chatbot, this chatbot is based on a FAQ, and the conversation turns are fairly independent from each other. So the main difficulty is again in classifying the user's intent based on the intents our model was exposed to at its conception. In this case as well, little data is initially available for the design of the chatbot.

This dataset initially contained 2 formulations for each of the 275 questions. A first chatbot was trained on this corpus. It is described as the reference experiment with classic StarSpace in Section 4.2.2. The chatbot was then exposed to about 30 of its users - the members of the legal services - at the NBC, in this preliminary version, to collect real interactions. The new formulations for existing intents made it possible to generate a test set that was representative of the users' actual questions. After filtering questions outside the FAQ framework and removing duplicates, we annotated 292 interactions covering 126 intentions to create the test dataset.

In this Section, we have presented two legal FAQs that will be the source of content for our chatbots. In the next Section, we will describe the design of these chatbots based on these small datasets. The first is intended to provide information on topics related to immigration to Canada. The second is integrated into a corporate framework to inform its users about the legal rules related to their work.

4. Experiments and results

4.1. Immigration chatbot

This Section presents our experiments to build an immigration chatbot. We start with the de-facto standard model in Rasa: the StarSpace intent classifier described in Section 2.4.3. Since our chatbot is based on FAQs, it is very similar to an IR task, so we compared this baseline with an IR-inspired model.

[8] https://www.cic.gc.ca/english/helpcentre/index-featured-can.asp
[9] https://gitlab.ikb.info.uqam.ca/marc/immigration_faq_scrapper

Table 4. Results of intent classification for the immigration chatbot

	microP	microR	microF$_1$	macroP	macroR	macroF$_1$	wP	wR	wF$_1$
softmax	0,92	0,52	0,67	0,51	0,52	0,52	0,51	0,52	0,52
marge	1	0,60	0,75	0,60	0,60	0,60	0,60	0,60	0,60
IR	1	0,60	0,75	0,60	0,60	0,60	0,60	0,60	0,60

4.1.1. Baseline using StarSpace

We trained a classifier on the immigration Q&A dataset, described in Section 3.2.1, using the StarSpace algorithm described in Section 2.4.3. We therefore created a common representation of user sentences and intents (one for example from the training set). The implementation of RASA[10] also returns a ranking of the instances with the highest similarity value to the input value.

We train two models for 40 epochs, meaning that the algorithm will see the entire dataset 40 times in total. The first model uses the standard loss described in Wu *et al.* [13], with the *mu* parameter, which controls the margin, at 0.8. For the second model, the loss is controlled by a function *softmax*, so there is no margin at which the training stops for a particular pair of instances. The function *softmax* takes as input a vector of K real numbers and produces a vector of K strictly positive real numbers whose sum is 1. Because of this feature, the output can be interpreted as a probability distribution. In the case of a multi-label classification, these probabilities are the chances that a given instance corresponds to each of the possible classes.

4.1.2. Approach Based on Information Retrieval

We also use a modified version of the StarSpace algorithm, which abandons the concept of intent to directly associate a user's sentence with an answer from the knowledge base. Rather than learning how to associate a text document to a class (which does not carry information), we bring two text documents together. We described in Section 2.4.3 how learning the position of entities is indirect. It is indeed the position of features (BoW) which is learned. The representation of the known words thus informs the learning of embeddings of other documents that use them, as in the original StarSpace algorithm. This approach, however, has two differences. First, adding the responses increases the amount of data available for learning the representations. Second, when predicting the correct response to a user interaction, the information contained in the response can be leveraged.

The first representation of documents uses n-grams with $n \in \{1, 4\}$, and then we train the embeddings for 40 epochs with the same parameters as the classic StarSpace model described in Section 4.1.1.

4.1.3. Results

The results of the experiments are reported in Table 4. Between the two versions of classic StarSpace, the model using loss with a margin outperforms the version using softmax. Note that in this case, using the StarSpace IR variant brings no improvement over the version which uses intents. There are several reasons for this. For example, the easy instances of the test corpus have already been classified correctly and the difficult instances cannot be classified by simple vocabulary similarity. Also, since the test covers only 88 of the 1088 classes, it may not be sufficiently representative of the entire domain for the model to be successfully applied. Variants using pre-trained models [19] might be more efficient but have the disadvantage of having more classes for fewer instances. The relative size of the questions and answers could also influence the results.

---

[10] https://github.com/RasaHQ/rasa

4.2. Legal information chatbot in a corporate environment

The following Section presents three systems providing NBC employees with answers to legal questions. We use StarSpace, and the IR variant previously described, and we compare these approaches with a system based on BERT [19], a transformer model.

4.2.1. Baseline using StarSpace

For the development of this chatbot, we also use the standard model provided by Rasa as a baseline, the StarSpace algorithm that we described in Section 2.4.3. This baseline model uses an initial representation of documents by n-grams with $n \in \{1, 2\}$. The model is run on 20 epochs with the default margin loss of StarSpace. The margin for positive instances (when to stop moving positive couples closer together) is $\text{mu}_{\text{pos}} = -0.8$ and the negative margin (when to stop moving negative couples further apart) is $\text{mu}_{\text{neg}} = -0.4$.

To understand why the value of mu_{neg} is negative, we have to go back to the definition of cosine similarity. If the word vectors being compared are BoW (vectors of binary values), the minimum value $\text{sim}_{\text{cosine}}$ can take is 0. However, for some characteristics such as the feeling associated with a term, negative values can make sense. Word embeddings, and those learned by StarSpace in particular, can learn such characteristics implicitly. In these cases, the lowest similarity that can be obtained is -1, which corresponds to a cos angle of π, or 180 degrees. This explains the negative value of mu_{neg}, since the range of possible values is actually [-1;1].

4.2.2. Enhanced Baseline

Another reference experiment to which we will compare our approach uses BERT [19] to represent conversations. BERT is a *transformer* (as described in 2.3) for which a pre-trained model is available. There is an English version, but no French version, so we use a trained model for 104 languages (including French). The pre-trained model is M-BERT$_{BASE}$[11], *i.e.* the basic version of BERT with 110M parameters, as opposed to the 345M parameters of $BERT_{LARGE}$ (which was not available in multilingual version). The model was then fine-tuned to the training dataset for 100 epochs, on a machine with a Nvidia GTX1070 graphics card. Using a model this large comes with additional constraints of compute power and inference time. Training this model from scratch is not realistic in most contexts, but fine-tuning a pre-trained one as we did is possible with relatively modest hardware.

4.2.3. Approach Based on Information Retrieval

Similar to the model described in Section 4.1.2, we use a classic StarSpace variant that uses the answers (to the questions in FAQ) to answer the questions.

Figure 6 illustrates the pipeline from questions in text form to their embedding representations using StarSpace. The text of the questions is pre-processed, and each word is then represented as an embedding. The representation of the question is the combination of the words that compose it. The representation of the intent ID is initialized at random, but it is then updated to be closer in the vector space to the words of the corresponding sentences.

In the IR-inspired version of StarSpace, questions are matched directly with answers, rather than having the intermediate steps of intent recognition (and optionally next action prediction). Figure 7 represents how the representation learning pipeline differs from the classic version of StarSpace. At the initialization, questions and answers are matched together, then pre-processed in the same way. All text documents, whether they are questions or answers, share the word embeddings. The embedding positions are updated for each question-answer pair, to be closer to that of words present in their

[11] https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip

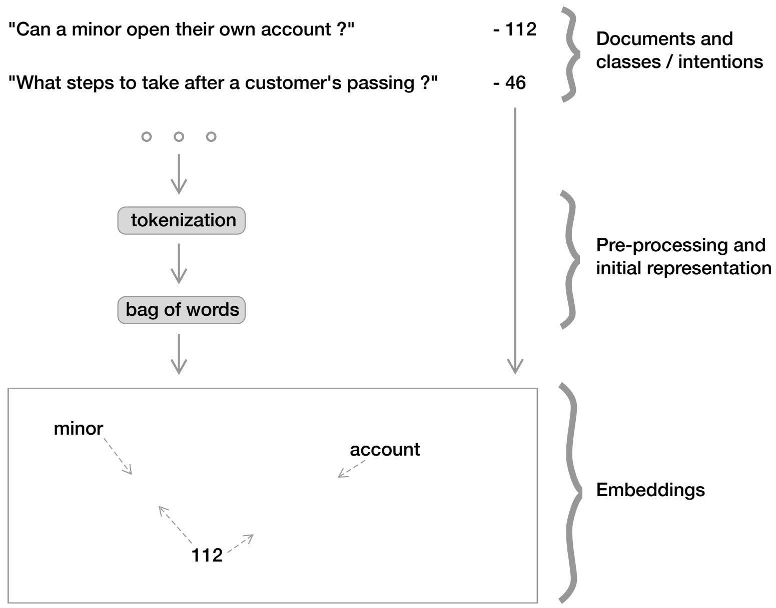


Figure 6. Embeddings learning pipeline for both questions and intents

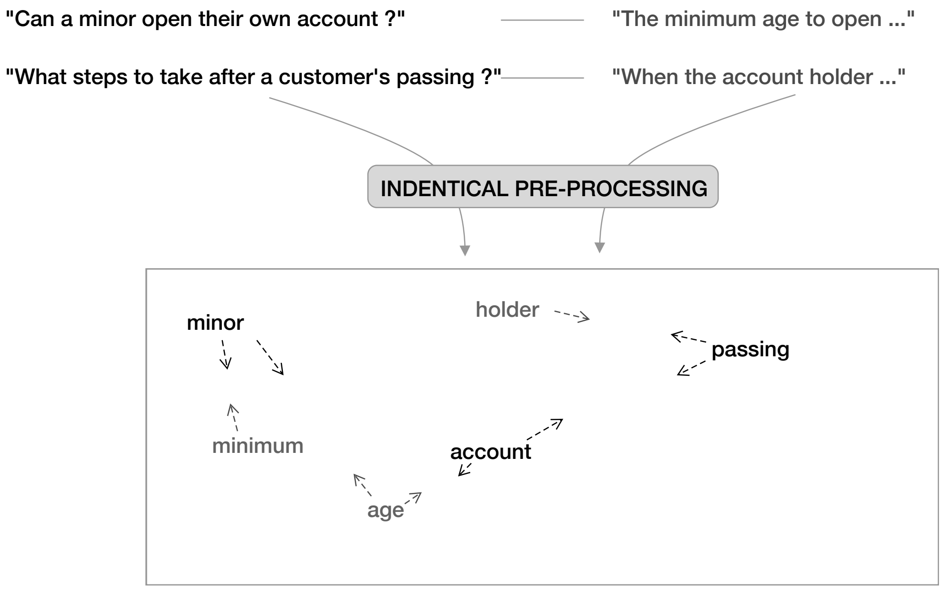


Figure 7. Embeddings learning pipeline for both questions and answers

match. For instance, the representations of "'minimum" and "minor" are updated to be closer to each other because they are are both used in the first question-answer pair of our example.

4.2.4. Results

Results of the legal data experiments are reported in Table 5. The standard version of StarSpace is outperformed by the other two approaches. Performance using an average of predictions at the instance level (microP and microF_1) gives our model the advantage, while for the other two types of averaging (macro and weighted), the BERT model is better. If we consider the differences between the original StarSpace model and the IR modification, we observe differences of more than 10% of F_1 in favor of the second one. It appears that the IR-based model uses the additional information contained

Table 5. Intention classification results on NBC legal chatbot

	microP	microR	microF$_1$	macroP	macroR	macroF$_1$	wP	wR	wF$_1$
StarSpace	0,61	0,61	0,61	0,52	0,48	0,48	0,73	0,61	0,63
BERT	0,70	0,67	0,66	0,75	0,75	**0,75**	0,85	0,75	**0,76**
IR	0,78	0,64	**0,70**	0,64	0,62	0,60	0,72	0,64	0,65

in the responses to better rank user interactions. However, this difference is not as large if we use a weighted average by the number of instances in each class, which could be similar to the questions presented in 4.1.2. The performance level of the IR-based model is also not sufficient to exceed the performance level of a language model like BERT, which takes advantage of information learned from the entire content of Wikipedia. While training the model from scratch is extremely cost-prohibitive, using a pre-trained model in a low-resource context is beneficial. Considering the weight of the model and especially the inference time, however, a simple IR-based model like the one we presented offers a significant improvement over a simple StarSpace model.

5. Conclusion

Access to legal information is a major obstacle to access to justice. In this article, we have designed two chatbots in order to inform their users about legal issues. One answers immigration-related questions, and the other, relying on a knowledge base of the NBC, answers legal questions from its employees. Both are based on FAQs, with the number of questions not exceeding, or barely exceeding, the number of answers. The underlying classification task therefore has a very low number of examples per class (less than 5) for a very high number of classes (275 and 1088 respectively). We have hence experimented with an algorithm to learn embeddings in a supervised way.

Faible qté de données et probablement déséquilibre entre les classes

I suggest for further work to check dataless and few-shot text classification litterature

We noticed that, in the least extreme case of the NBC dataset, the use of a StarSpace variant, which represents the answers in the same space and uses them to make predictions, brings a very significant improvement. On the same dataset, using a language model pre-trained on a very large corpus and fine-tuned to ours further increases performance. The extra training cost is unrealistic for all but the biggest system providers in the field, but the existence of pre-trained networks on some languages limits the impact of this drawback. On the immigration dataset, the variant does not bring any performance improvement. These results should be put into perspective with the small size of the test dataset compared to the training dataset (only 10% of the classes are covered by the test dataset).

cross-validation or better one held-out

Both chatbots we described consisted mainly in an intent recognition module. The NBC chatbot also uses keyword-based query disambiguation [44] techniques to increase its overall score. We plan to supplement our immigration chatbot with documents from case law, collected and annotated in a previous work [45]. While the information these documents contain would not be as useful to the average reader as the content specifically targeted at them, the added context would help where no clear-cut answer is provided. This would also serve as an additional step in a proof-of-concept for an information system that would eventually support lawyers. More information about these use cases can be found in the LegalIA presentation video[12].

https://gitlab.ikb.info.uqam.ca/marc/immigration_bot https://gitlab.ikb.info.uqam.ca/marc/immigration_faq_scrapper

Supplementary Materials: The source code related to the immigration chatbot is available under MIT licence to ensure reproducibility. It can be found in these two repositories: https://gitlab.ikb.info.uqam.ca/ marc ~~legalia~~/immigration_bot(chatbot) and https://gitlab.ikb.info.uqam.ca/ marc ~~legalia~~/immigration_faq_scrapper (data collection). The repositories are not accessible freely, because one need to sing in first.

Author Contributions: Conceptualization, M.Q. and M.-J.M.; methodology, M.Q. and M.-J.M.; software, M.Q.; validation, M.Q., E.C. and M.-J.M.; investigation, M.Q., E.C. and M.-J.M.; resources, E.C. and M.-J.M.; data curation, M.Q.; writing–original draft preparation, M.Q.; writing–review and editing, M.-J.M.; supervision, M.-J.M. All authors have read and agreed to the published version of the manuscript.

[12] https://legalia.uqam.ca/video/LegalIA.mp4 this link doesn't work

Funding: This research was funded by [M.-J.M.] FRQ Audace Grant LegalIA (2018) and Canada NSERC Grant number 06487-2017. NBC funded an intership for M.Q.

Acknowledgments: The authors want to thank Me Stefanny Beaudoin for her support, guidance and help in understanding the legal domain. They also thank Arcady Gascon-Afriat for carefully proofreading this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

BERT Bidirectionnal Encoder Representations from Transformers. 7, 15, 16, 17
BoW Bag-of-Words. 4, 5, 8, 14, 15

CNN Convolutional Neural Network. 7

FAQ Frequently Asked Questions. 11, 13, 15, 17

IR Information Retrieval. 2, 8, 10, 13, 14, 15, 16, 17

ML Machine Learning. 11

NBC National Bank of Canada. 2, 8, 13, 15, 17, 18
NLP Natural Language Processing. 2, 4, 5, 7, 8, 9
NRLs Non-Represented Litigants. 1

QAS Question Answering System. 7, 8

RNN Recurrent Neural Network. 6, 7, 10

TF-IDF Term Frequency-Inverse Document Frequency. 5

UQAM Université du Québec à Montréal. 1, 2

References

1. Laniel, R.A.; Bahary-Dionne, A.; Bernheim, E. Agir seul en justice: du droit au choix—État de la jurisprudence sur les droits des justiciables non représentés. *Les Cahiers de droit* **2018**, *59*, 495–532.
2. Schneider, R.K. Illiberal Construction of Pro Se Pleadings. *University of Pennsylvania Law Review* **2010**, *159*, 585.
3. Manning, C.D.; Manning, C.D.; Schütze, H. *Foundations of Statistical Natural Language Processing*; MIT press, 1999.
4. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: New York, NY, USA, 2008.
5. Spärck Jones, K. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of documentation* **1972**, *28*, 11–21.
6. Shannon, C.E.; Weaver, W. *A Mathematical Theory of Communication*; University of Illinois press, 1998.
7. Robertson, S. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation* **2004**, *60*, 503–520.
8. Harris, Z.S. Distributional Structure. *WORD* **1954**, *10*, 146–162. doi:10.1080/00437956.1954.11659520.
9. Landauer, T.K.; Dumais, S.T. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological review* **1997**, *104*, 211.
10. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781* **2013**.
11. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
12. Arora, S.; Liang, Y.; Ma, T. A Simple but Tough-To-Beat Baseline for Sentence Embeddings, 2016.
13. Wu, L.Y.; Fisch, A.; Chopra, S.; Adams, K.; Bordes, A.; Weston, J. Starspace: Embed all the things! Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

14. Kenter, T.; Borisov, A.; De Rijke, M. Siamese CBOW: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640* **2016**.
15. Xu, W.; Rudnicky, A. Can Artificial Neural Networks Learn Language Models? Sixth International Conference on Spoken Language Processing, 2000.
16. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A Neural Probabilistic Language Model. *Journal of machine learning research* **2003**, *3*, 1137–1155.
17. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078* **2014**.
18. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473* **2014**.
19. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* **2018**.
20. Jurafsky, D.; Martin, J.H. Question Answering. In *Speech and Language Processing*; Pearson, 2014; Vol. 3.
21. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A Nucleus for a Web of Open Data. Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference; Springer-Verlag: Berlin, Heidelberg, 2007; ISWC'07/ASWC'07, p. 722–735.
22. Deriu, J.; Rodrigo, A.; Otegi, A.; Echegoyen, G.; Rosset, S.; Agirre, E.; Cieliebak, M. Survey on Evaluation Methods for Dialogue Systems. *arXiv preprint arXiv:1905.04071* **2019**.
23. Weizenbaum, J.; others. ELIZA—a Computer Program for the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM* **1966**, *9*, 36–45.
24. Wallace, R.; Tomabechi, H.; Aimless, D. Chatterbots Go Native: Considerations for an Eco-System Fostering the Development of Artificial Life Forms in a Human World. *Published online: http://www. pandorabots. com/pandora/pics/chatterbotsgonative. doc* **2003**.
25. Satu, M.S.; Parvez, M.H.; others. Review of Integrated Applications With AIML Based Chatbot. 2015 International Conference on Computer and Information Engineering (ICCIE). IEEE, 2015, pp. 87–90.
26. Vinyals, O.; Le, Q. A Neural Conversational Model. *arXiv preprint arXiv:1506.05869* **2015**.
27. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
28. Qiu, M.; Li, F.L.; Wang, S.; Gao, X.; Chen, Y.; Zhao, W.; Chen, H.; Huang, J.; Chu, W. AliMe Chat: A Sequence to Sequence and Rerank Based Chatbot Engine. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017, Vol. 2, pp. 498–503.
29. Turing, A.M. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433.
30. Radziwill, N.M.; Benton, M.C. Evaluating Quality of Chatbots and Intelligent Conversational Agents. *arXiv preprint arXiv:1704.04579* **2017**.
31. Ramos, R. Screw the Turing Test-Chatbots don't need to act human. *VentureBeat. Retrieved on March* **2017**, *13*, 2017.
32. Deriu, J.; Rodrigo, A.M.; Otegi, A.; Echegoyen, G.; Rosset, S.; Agirre, E.; Cieliebak, M. Survey on Evaluation Methods for Dialogue Systems. *ArXiv* **2019**, *abs/1905.04071*.
33. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002, pp. 311–318.
34. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
35. Lowe, R.; Noseworthy, M.; Serban, I.V.; Angelard-Gontier, N.; Bengio, Y.; Pineau, J. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. *arXiv preprint arXiv:1708.07149* **2017**.
36. Bobrow, D.G.; Kaplan, R.M.; Kay, M.; Norman, D.A.; Thompson, H.; Winograd, T. GUS, a Frame-Driven Dialog System. *Artificial intelligence* **1977**, *8*, 155–173.
37. Meurs, M.J.; Duvert, F.; Lefevre, F.; De Mori, R. Markov Logic Networks for Spoken Language Interpretation. *Information Systems Journal* **2008**, *1978*, 535–544.
38. Meurs, M.J.; Lefevre, F.; De Mori, R. Learning Bayesian Networks for Semantic Frame Composition in a Spoken Dialog System. Proceedings of Human Language Technologies: The 2009 Annual Conference of

the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, 2009, pp. 61–64.
39. Jurafsky, D.; Martin, J.H. Dialog Systems and Chatbots. *Speech and language processing* **2017**, *3*.
40. Bocklisch, T.; Faulkner, J.; Pawlowski, N.; Nichol, A. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181* **2017**.
41. Baker, M. 1,500 Scientists Lift the Lid on Reproducibility. *Nature News* **2016**, *533*, 452.
42. Gundersen, O.E.; Kjensmo, S. State of the art: Reproducibility in artificial intelligence. Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
43. Zaharia, M.; Chen, A.; Davidson, A.; Ghodsi, A.; Hong, S.A.; Konwinski, A.; Murching, S.; Nykodym, T.; Ogilvie, P.; Parkhe, M.; others. Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.* **2018**, *41*, 39–45.
44. Charton, E.; Meurs, M.J.; Jean-Louis, L.; Gagnon, M. Mutual Disambiguation for Entity Linking. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 476–481.
45. Queudot, M.; Meurs, M.J. Artificial Intelligence and Predictive Justice: Limitations and Perspectives. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, 2018, pp. 889–897.