

This pdf file encompasses the figures and images produced in the notebook, a.k.a., all the neat results, no code included.

Background

Freedom of Information Requests in the Region of Waterloo

One of the Region of Waterloo initiatives is [Open Data](#). With it, the Region strives to be open, transparent, and accountable to citizens. It shares its data for everyone to use and republish with few restrictions. The data is provided in machine-readable format.

Searching the Region's Open Data Portal, one can find the Freedom of Information Requests (FOIR) data set. This data set spans 18 years (1999-2016).

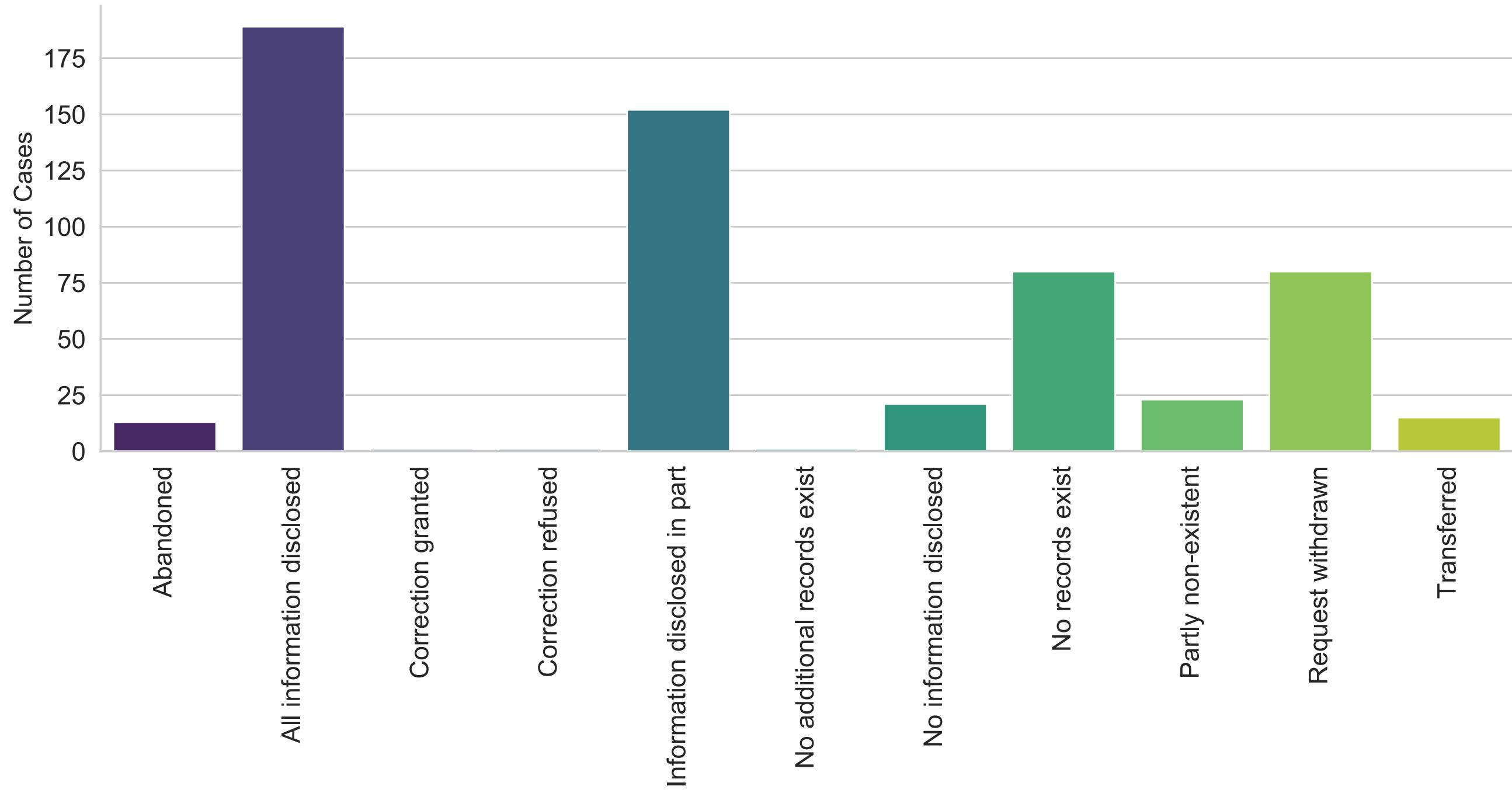
In the repository, you will find a jupyter [notebook](#) that does a thorough job at describing the FOIR set. Specially from the side of descriptive statistics, visualization, natural language processing (NLP), and topic modeling (LSA, LDA, LSI).

Overview of Notebook

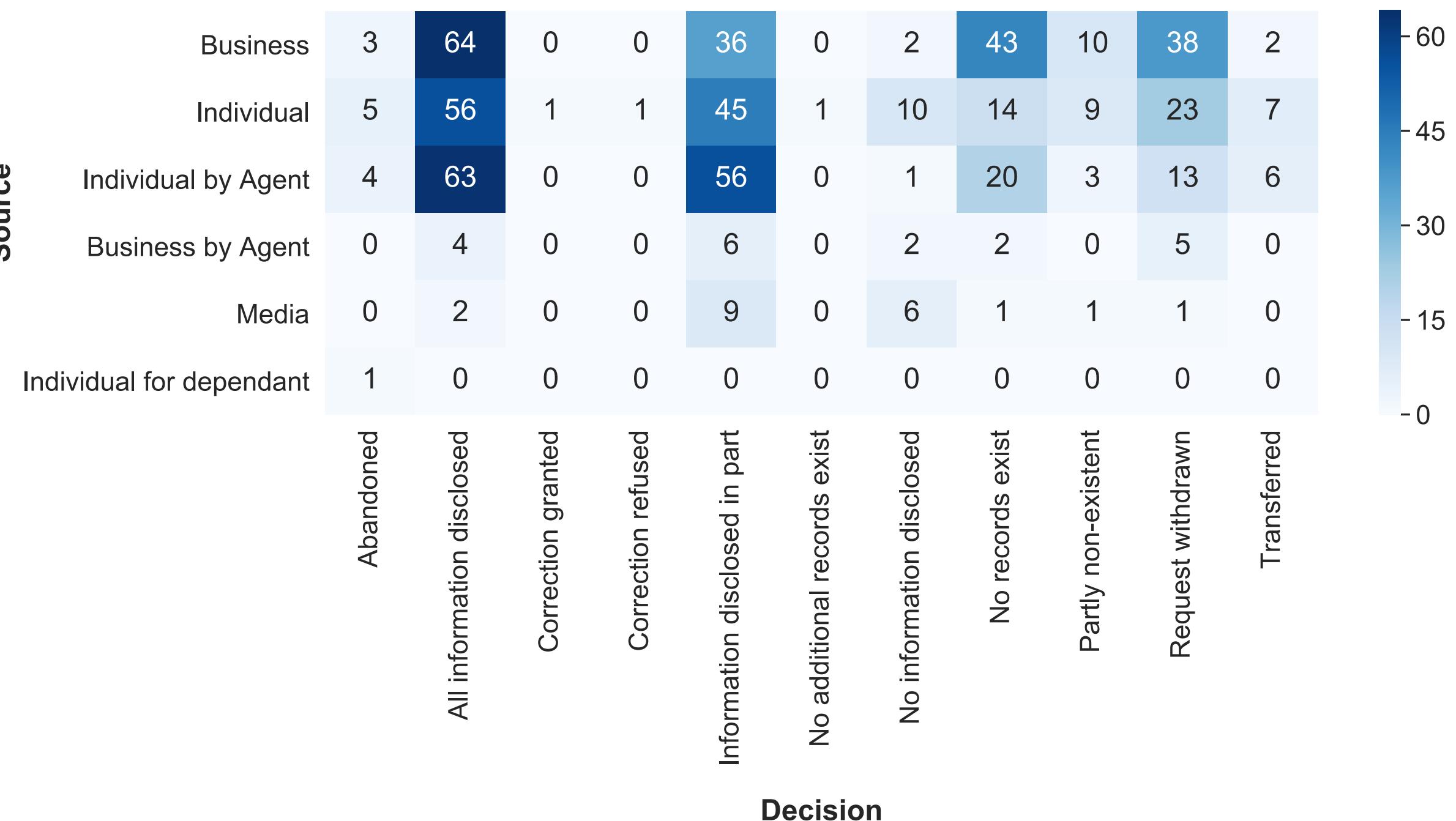
- Get to know the data
 - We look at the files, we merge them, and clean them if necessary.
 - We have five main columns: one is the request ID, three columns have categorical data, and one has the summary of the request itself (plain text).
 - Of those with categorical data, decision made, source (a.k.a., requester), and request type, can we somehow reduce the number of categories (spoiler alert: yes).
- Descriptive Analysis
 - There are about 11 types of decisions for six different types of sources.
 - How many requests per type of decision?
 - How are the decisions split based on the source? Are some sources more "lucky" than others? Spoiler alert: yes, though we don't know the reason.
 - How is each decision split among the sources?
 - For the main types of decisions (All information disclosed, Information disclosed in part, No information disclosed), how are they split among the sources?
- Natural Language Processing (NLP) - Analyzing the summary of request

- Before analyzing any text with NLP, one needs to go over some steps, which can vary depending on the goal:
 - Parse text, tokenize it, remove symbols, remove stopwords, remove punctuation, convert tokens to lowercase, remove short words, and lemmatize the tokens. We use both NLTK and spaCy.
- n-grams. We take a look at the most frequent unigrams, bigrams, trigrams, and n-grams (4-5).
- WordClouds. They are not only pretty, they are also useful. Why is a word/phrase so important!?
- Let's take six of these frequent phrases and find if there is any correlation with the decision taken.
- Summary of request statistics. With how much text are we working? How long are the requests before and after tokenization? (Spoiler alert: seven tokens is the median per request. Ouch!)
- Part-of-Speech (POS) tagging of the requests. Nouns, verbs?
- Topic Modeling
 - Here we do LSA and LDA Analysis using Bokeh, scikit-learn, and t-SNE.
 - We also try LDA Analysis using Gensim and pyLDAvis.
- Machine Learning (ML)
 - Here we compare the accuracies of eight classifiers, RandomForest, LinearSVC, MultinomialNB, LogisticRegression, SVC, KNeighbors, SGDClassifier, and DecisionTree, using pipelines, GridSearchCV, confusion matrices, and classification reports.
 - We compare two vectorizers, Count vectorizer and tf-idf vectorizer.
 - Given that some of our decisions have less than 15 instances and that we also have an unbalanced case, we look at other ways to optimize this. For example, a) we keep decisions with over 15 instances, b) we merge our 11 types of decisions into three main bins (full, partial, or no info released), and c) we remove cases where no decision was made (withdrawn or abandoned, which we name it as the independent case).
 - And with all of this, our best score goes up to... 51%

Decisions Made for all Requests

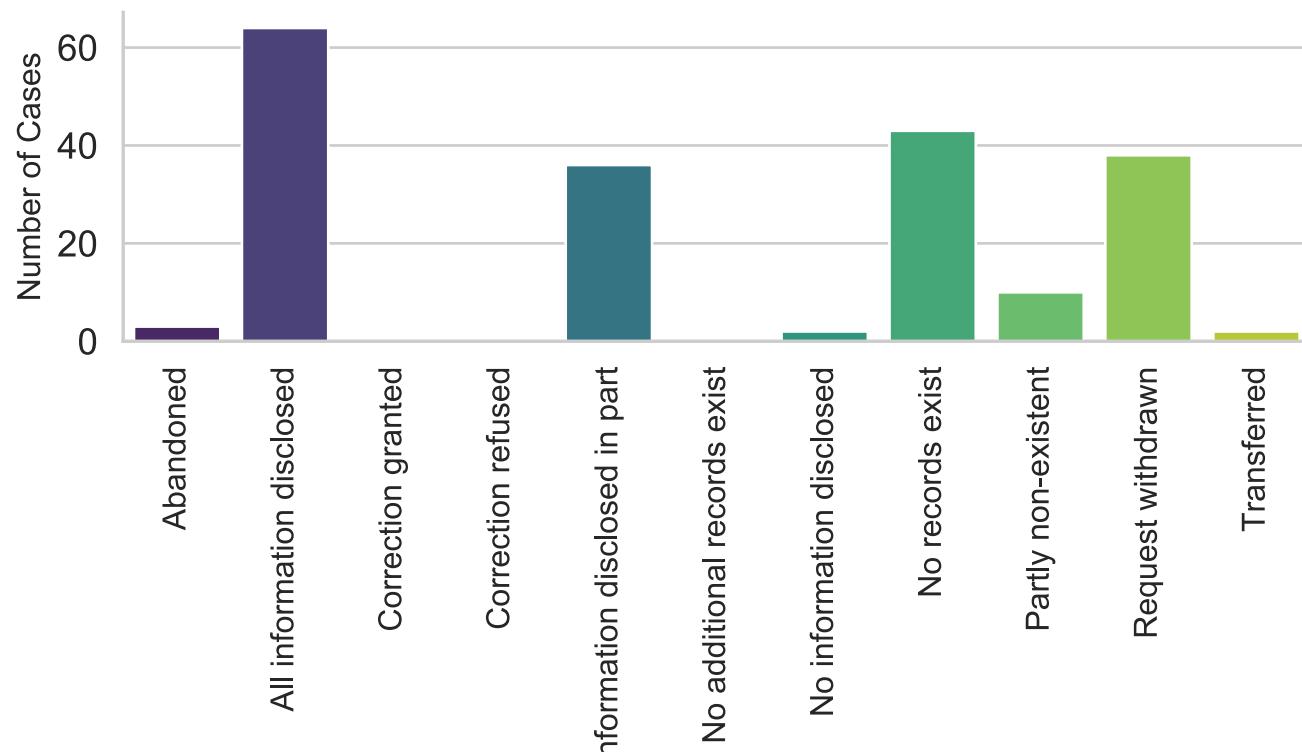


Full Data

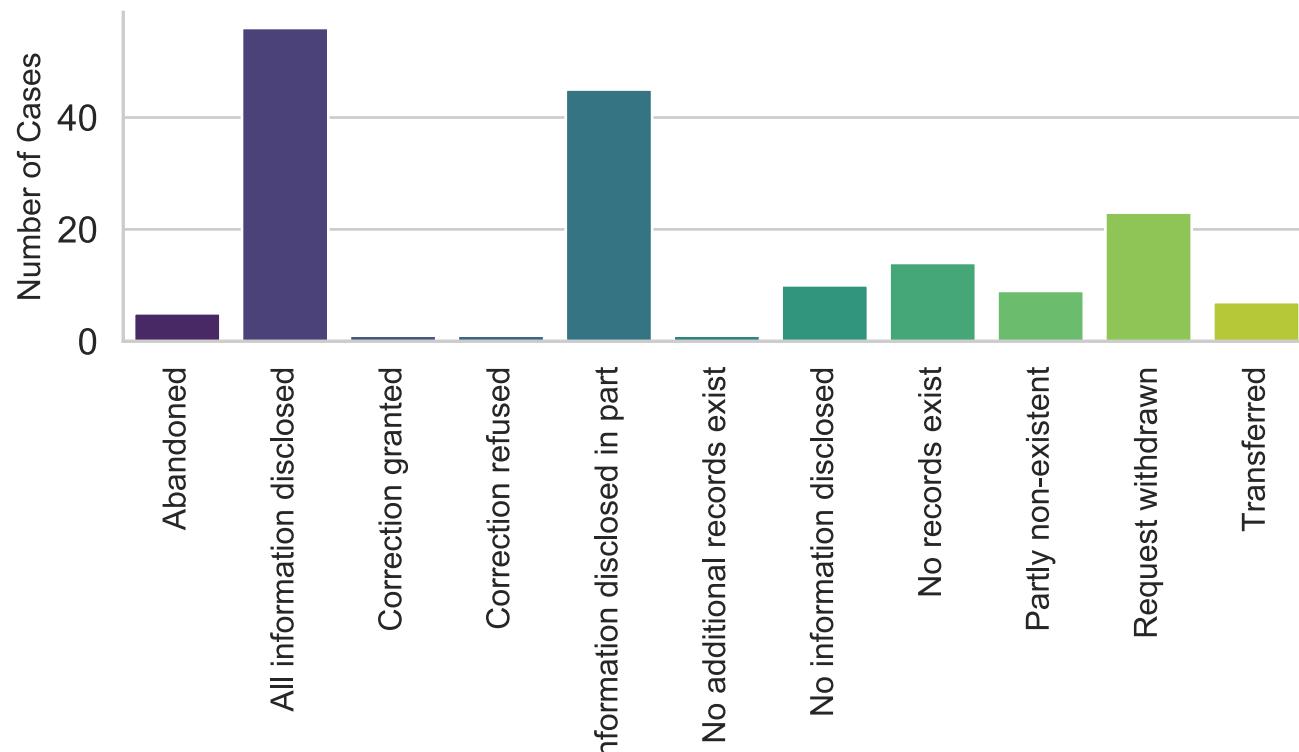


Number of cases for all type of decisions made for each of the sources

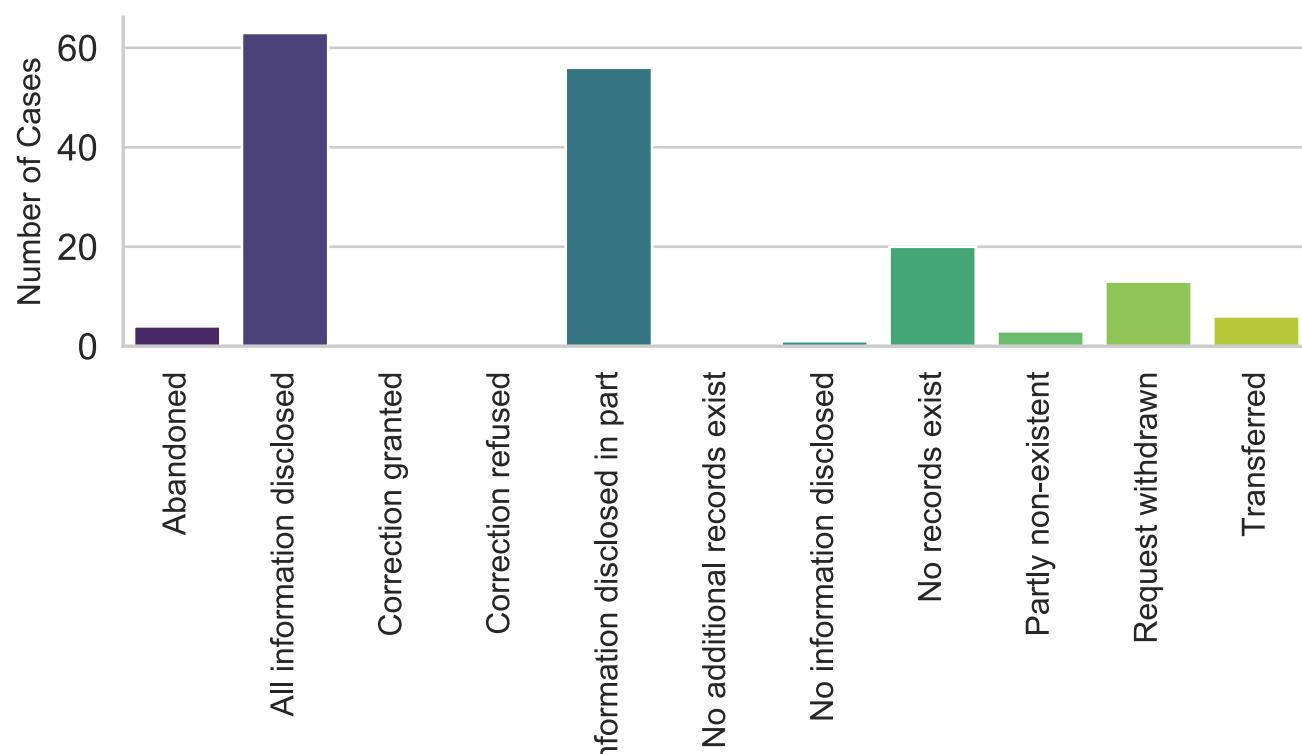
Requests made by 'Business'



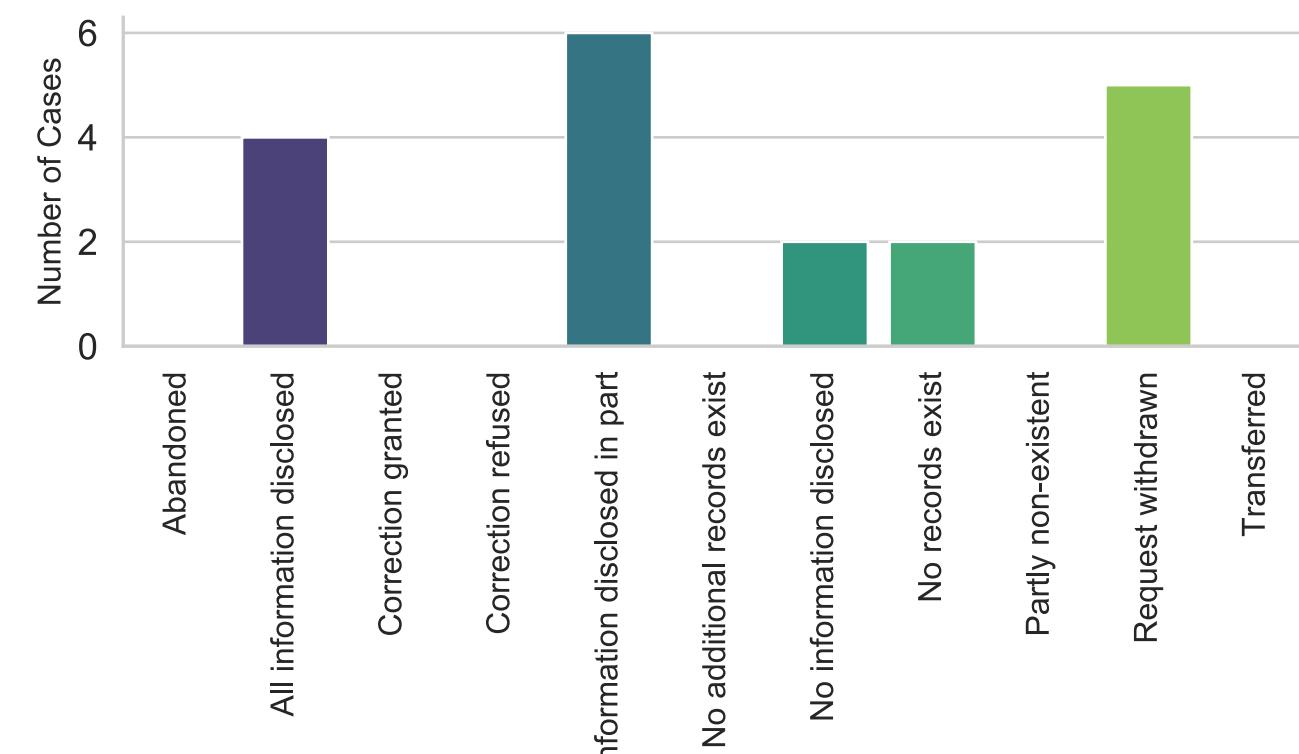
Requests made by 'Individual'



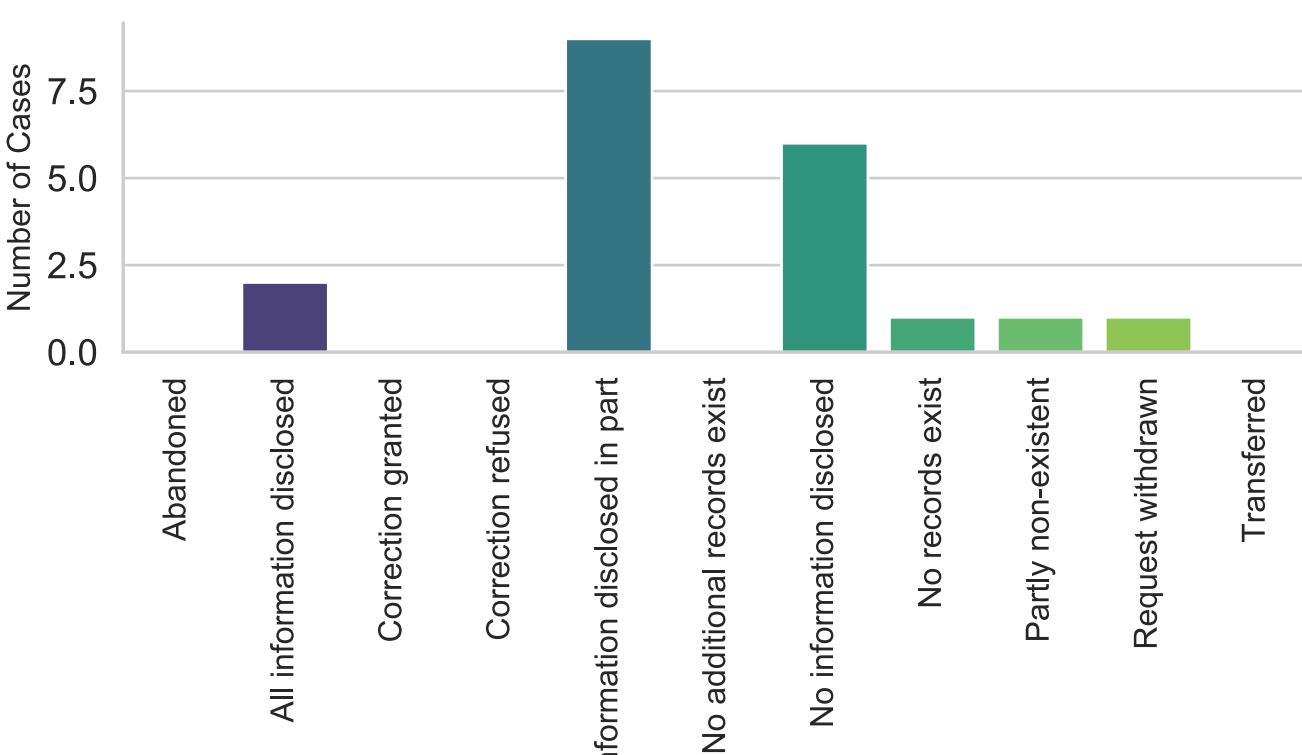
Requests made by 'Individual by Agent'



Requests made by 'Business by Agent'



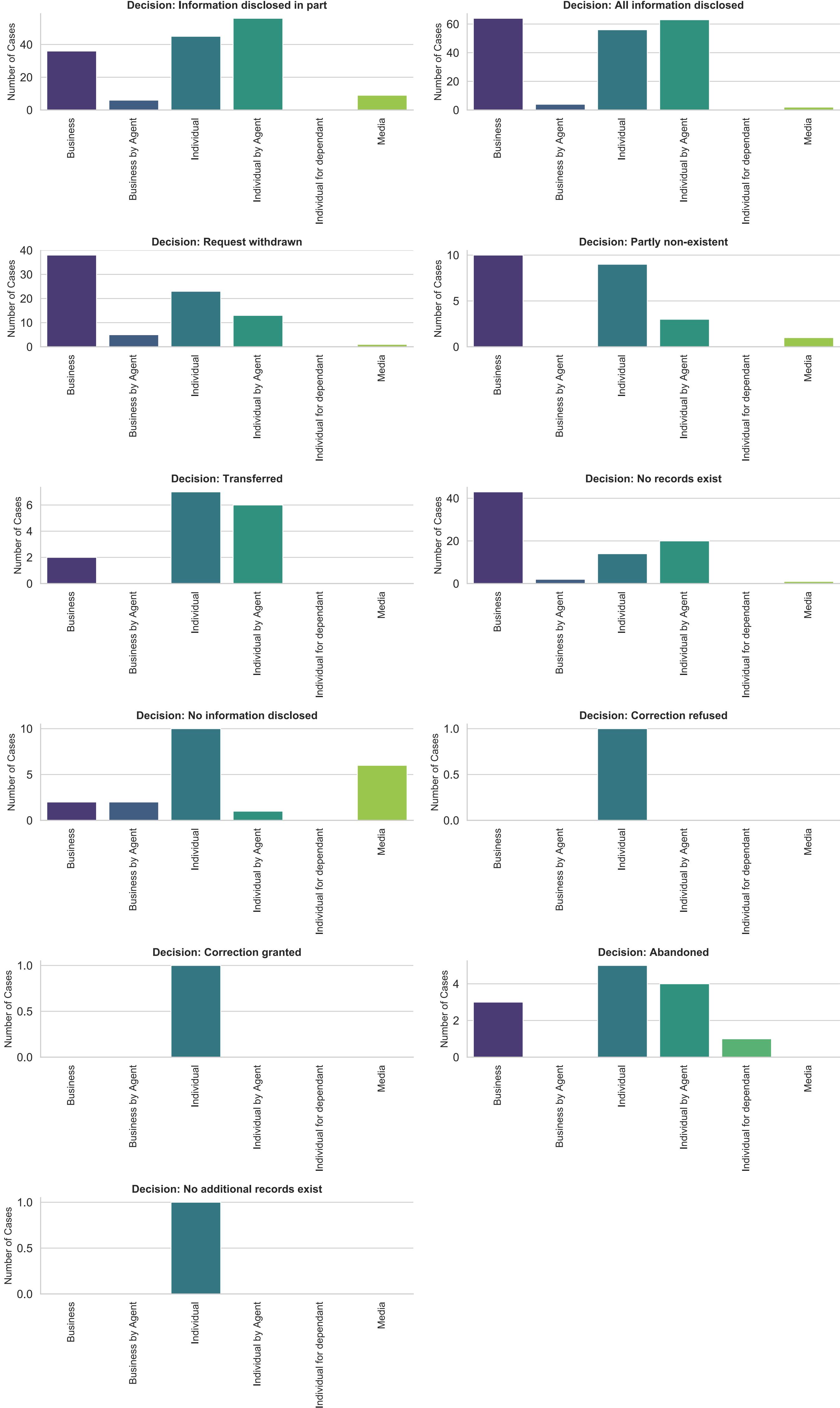
Requests made by 'Media'



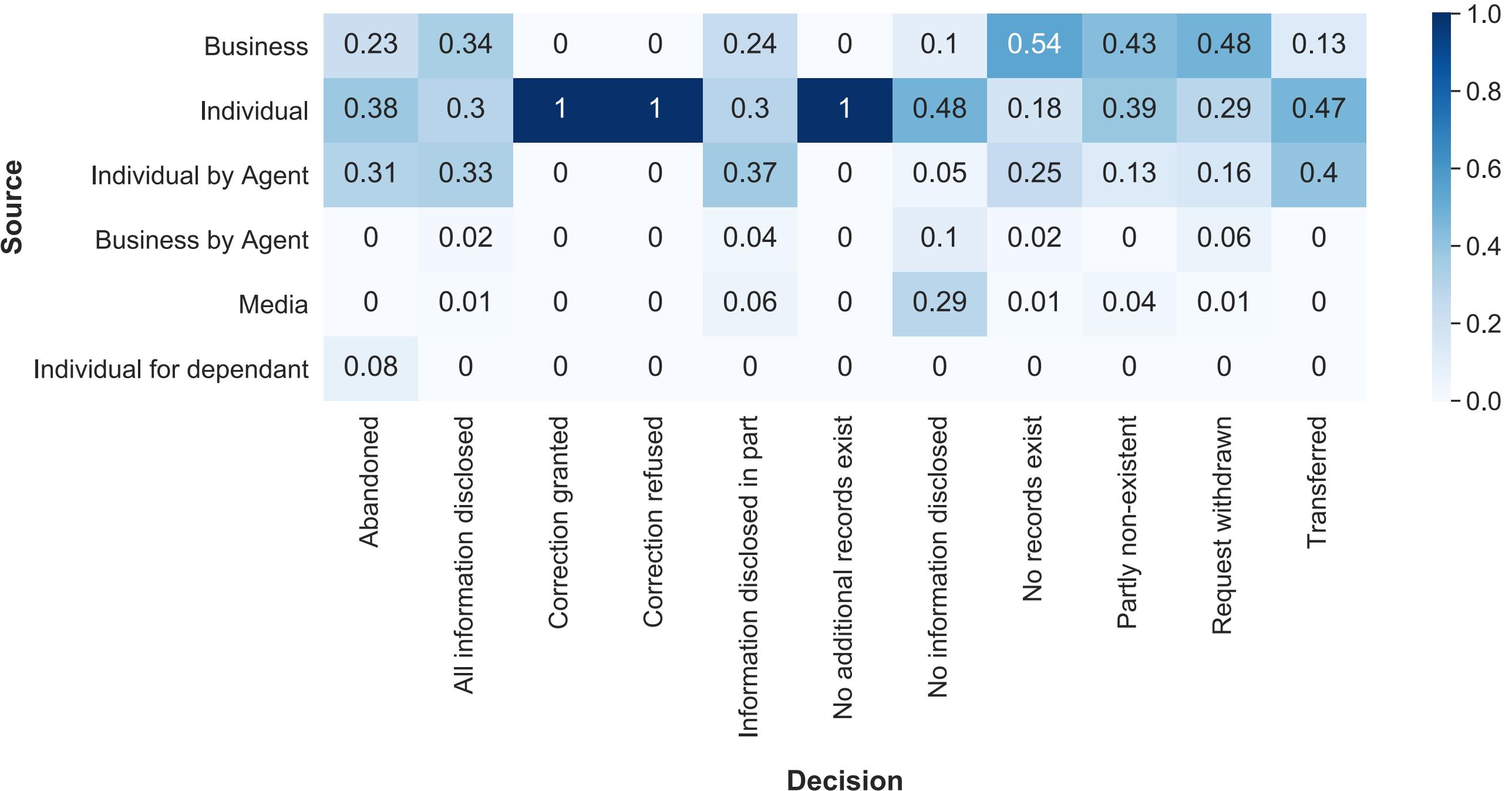
Requests made by 'Individual for dependant'



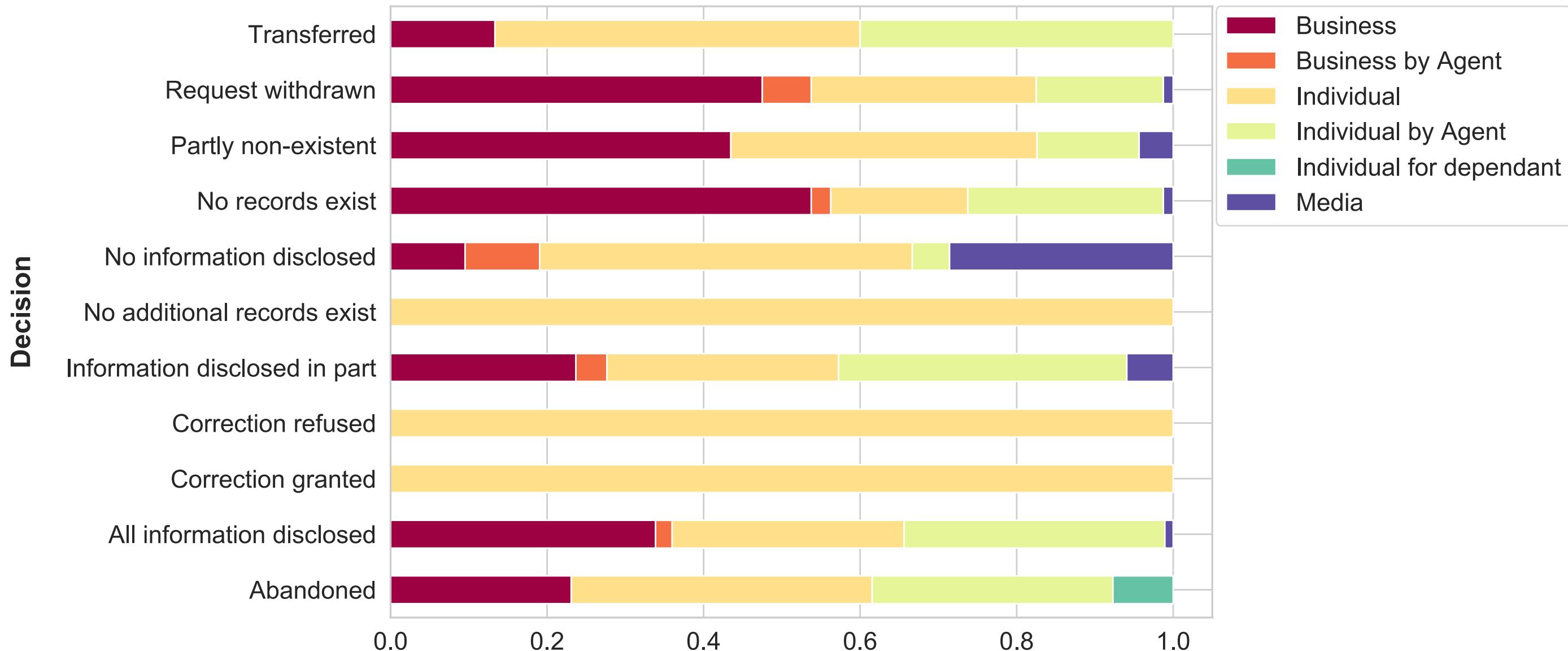
Number of cases for each type of decision made by sources



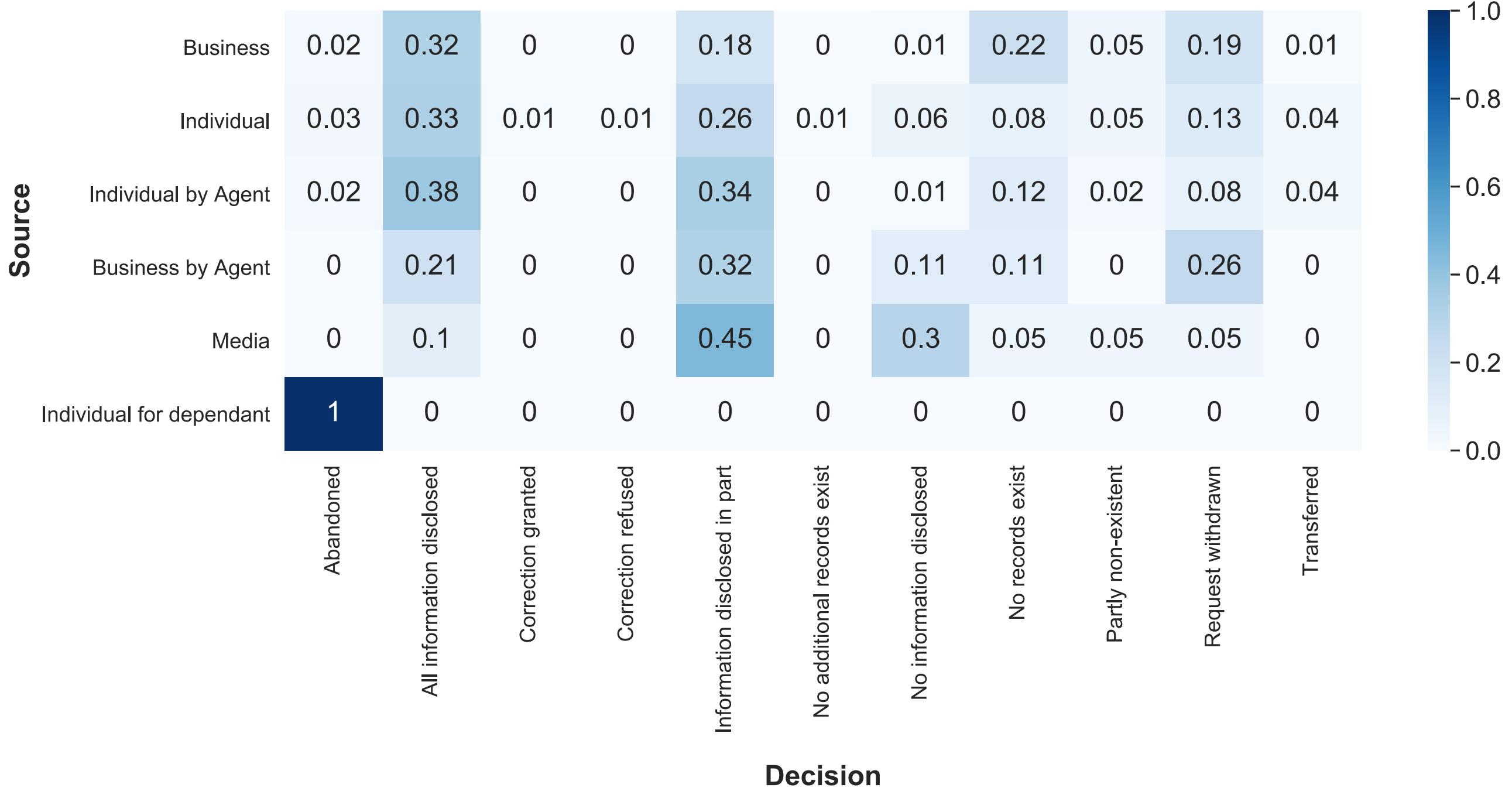
How each decision is split among all sources (fraction)



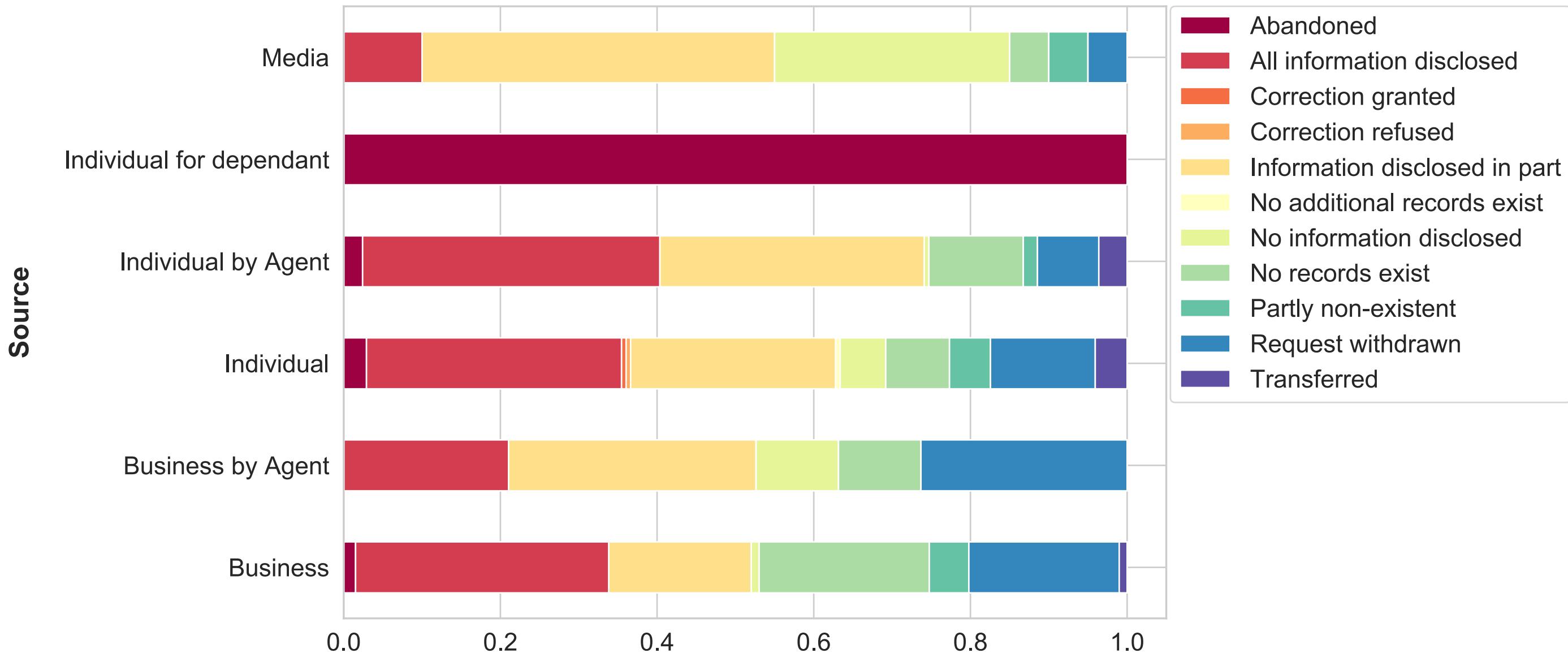
Full data, how each decision is split per source



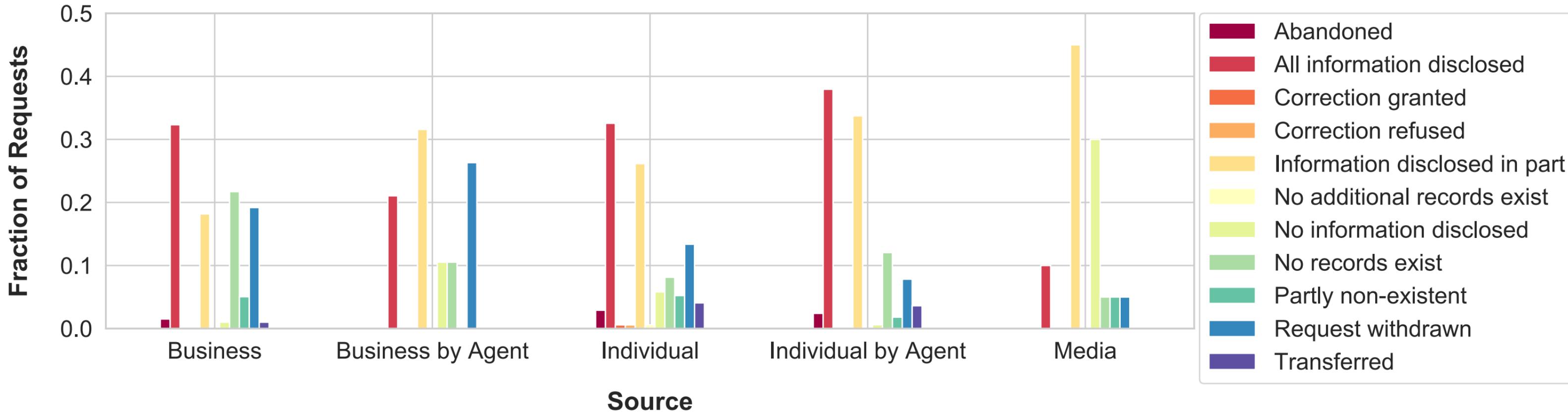
How decisions are split per source (fraction)



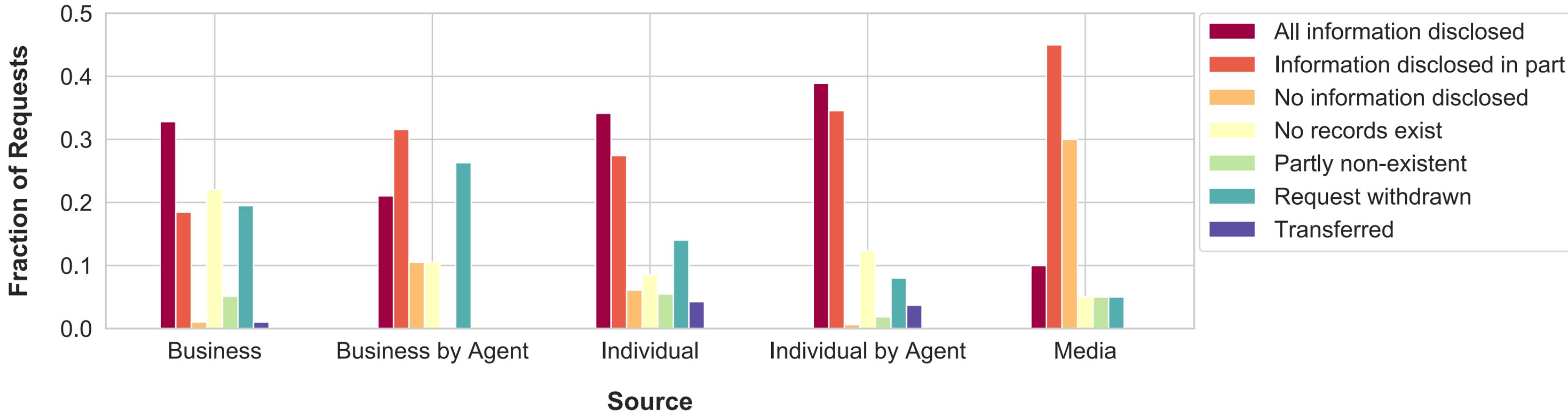
Full data, fraction of decisions per source



Full data, fraction of decisions per source



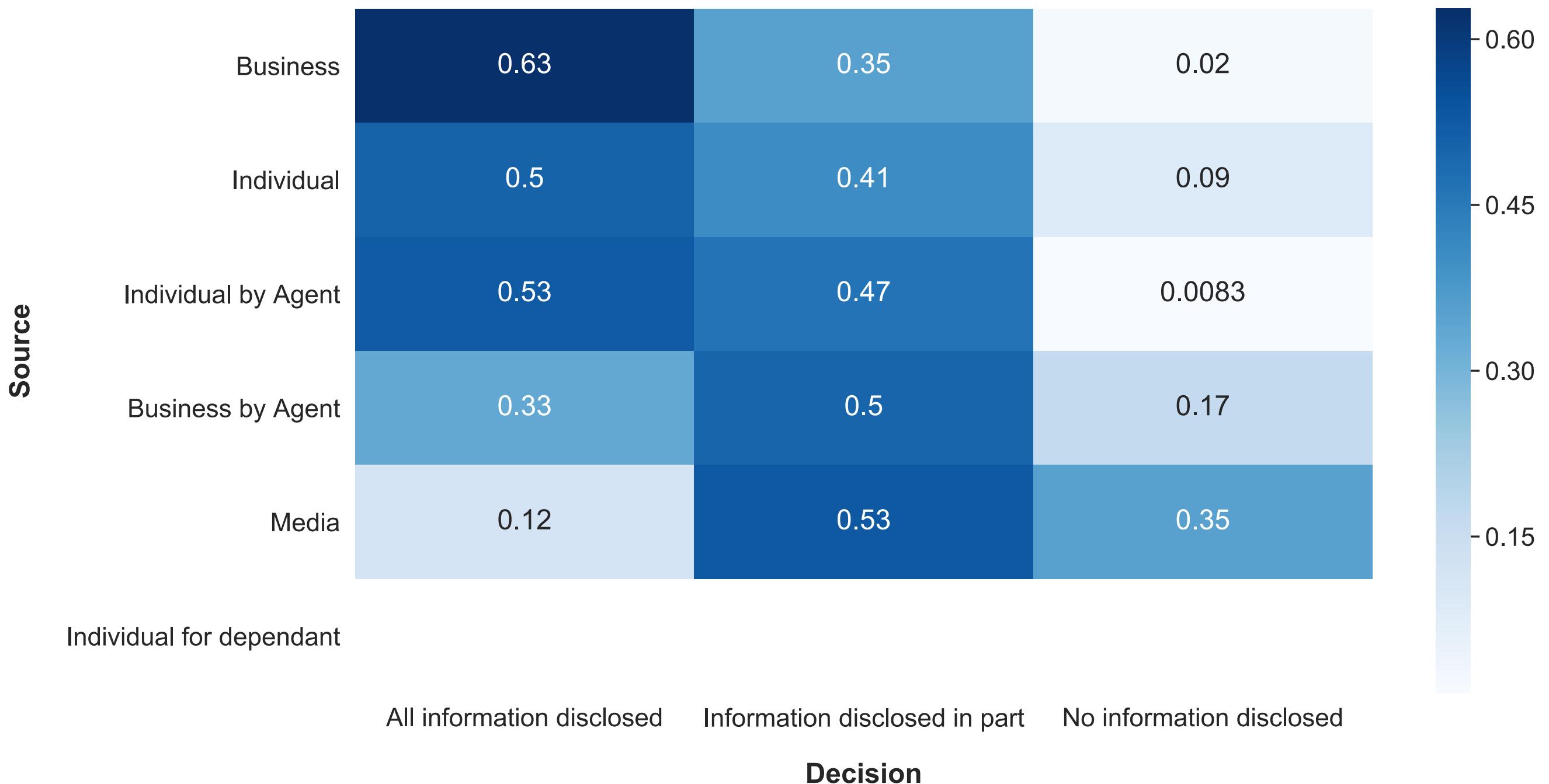
Fraction of decisions per source, for decisions with more than 15 instances only



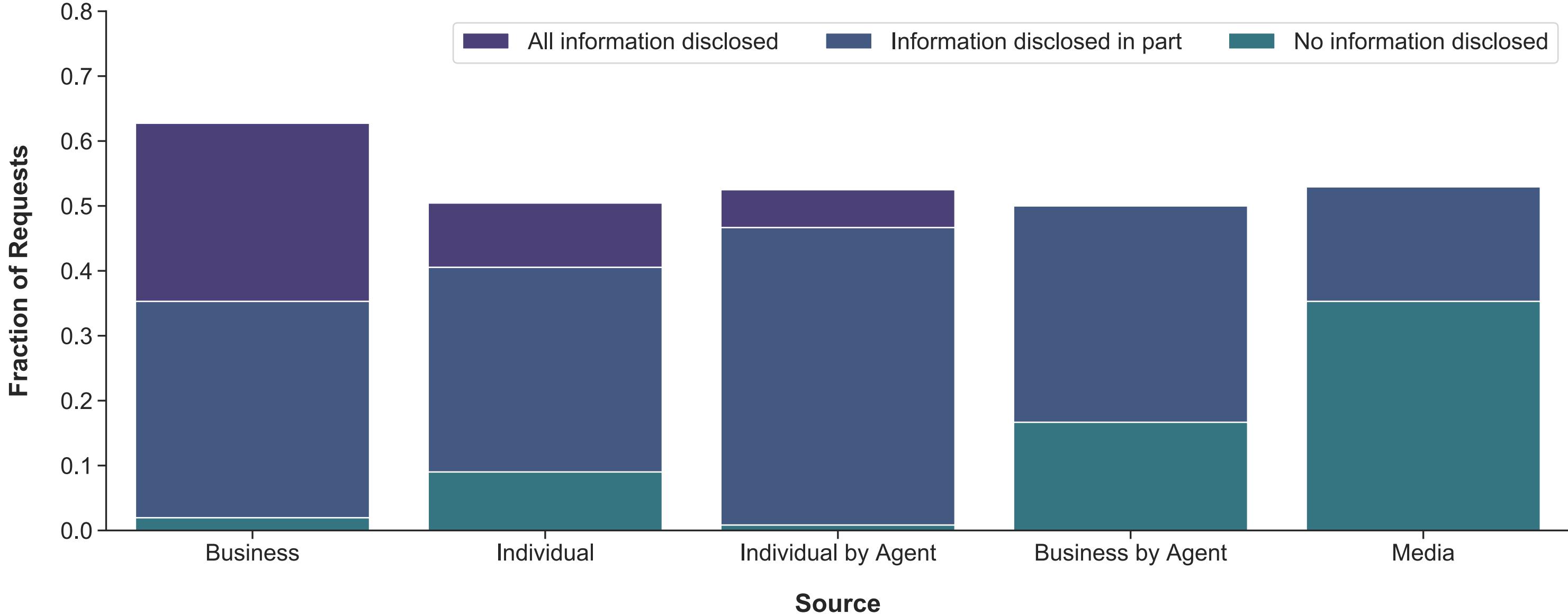
Each of the three main decisions split among all the sources (fraction)



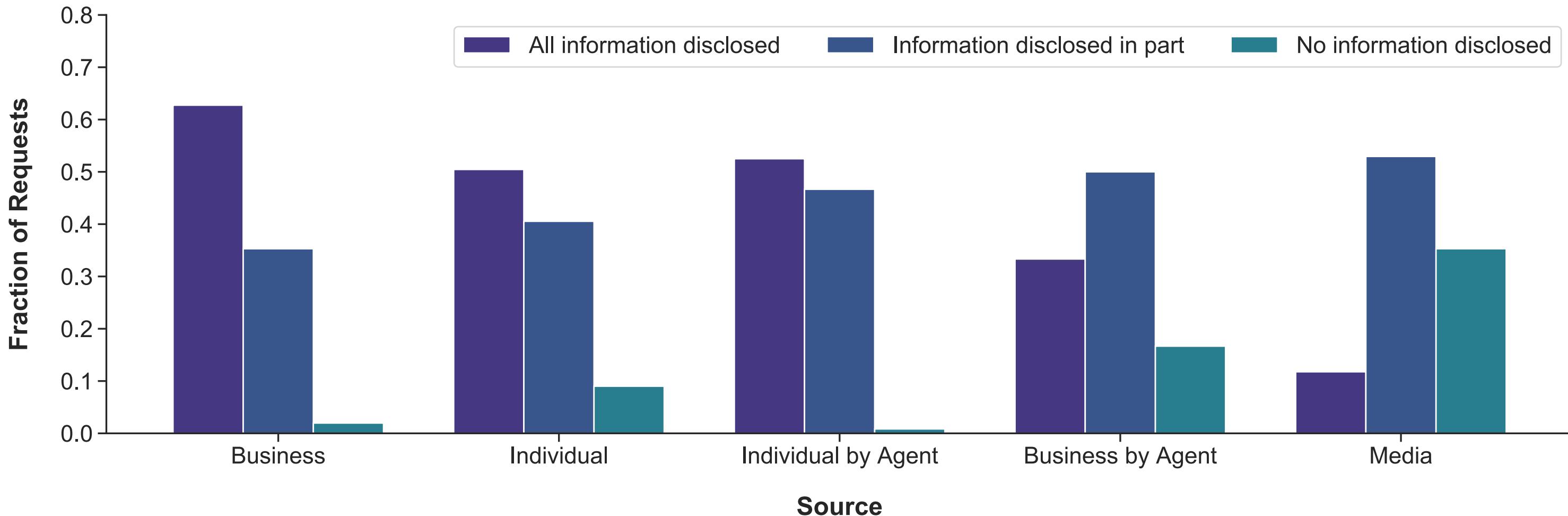
Three main decisions split for each source (fraction)



Three main decisions only, fractions for each source add to 1



Three main decisions only, fractions for each source add to 1



Top 200 unigrams/bigrams, full text



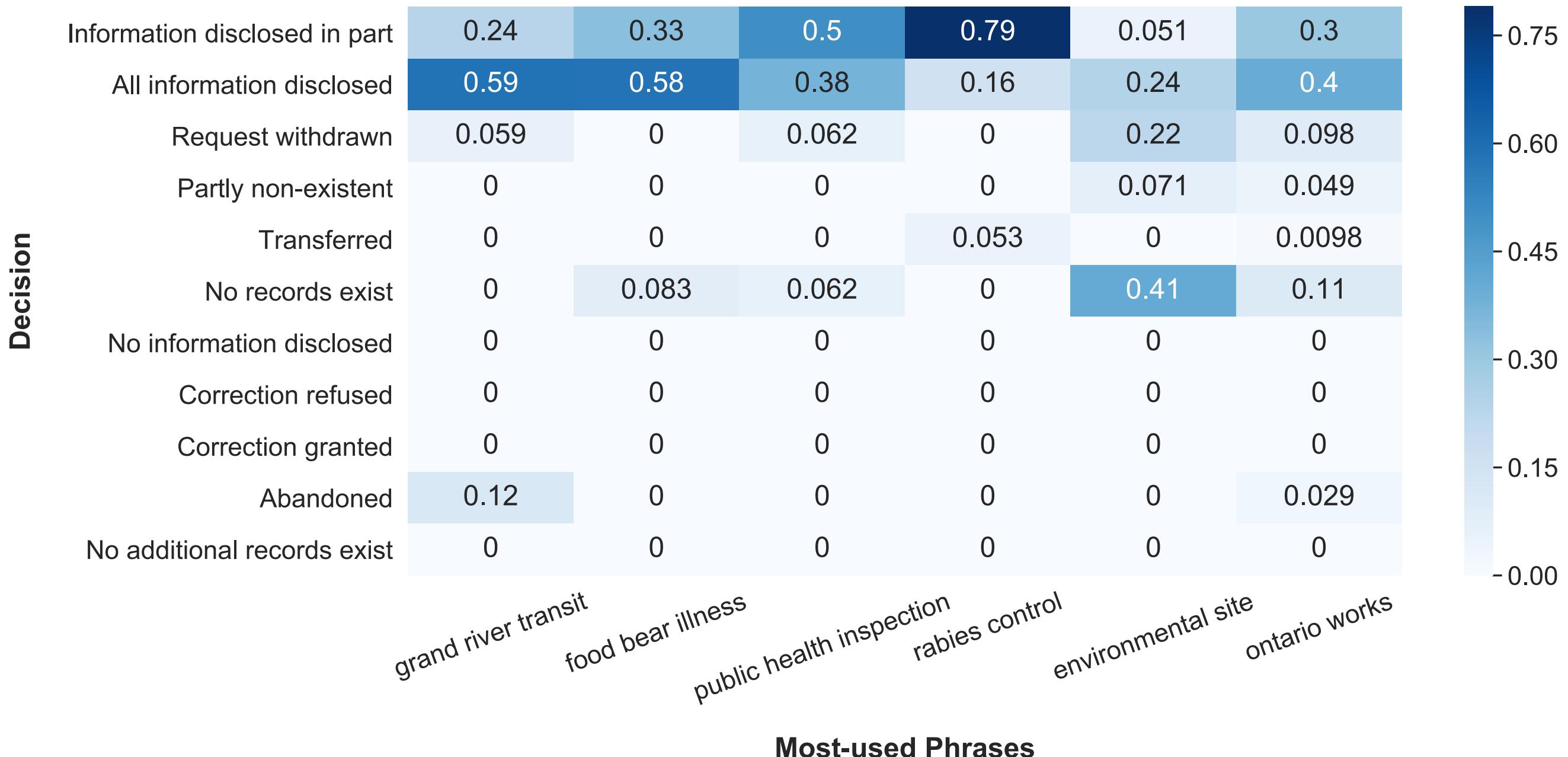
Top 200 unigrams, full text



Top 200 unigrams/bigrams, full text without '{* remove}'

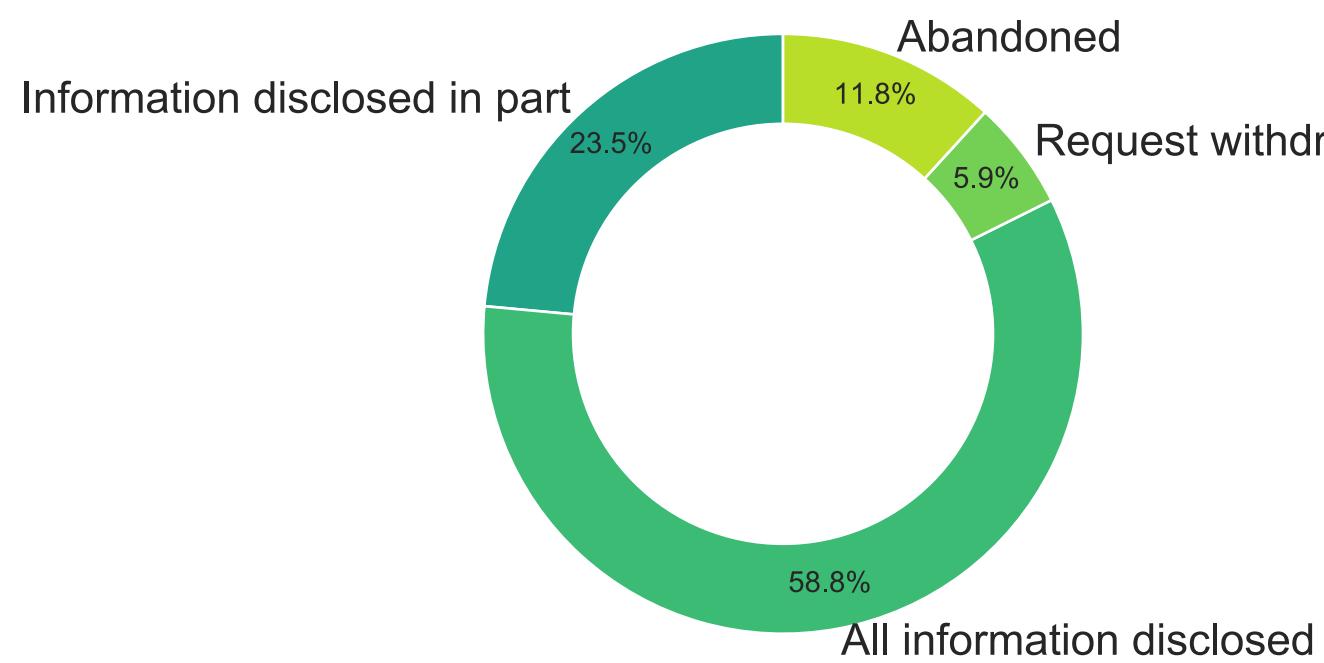
Top 200 unigrams, full text without '{* remove}'

46% of the full data uses the following phrases.
For each phrase, here is how decisions are split (fraction).

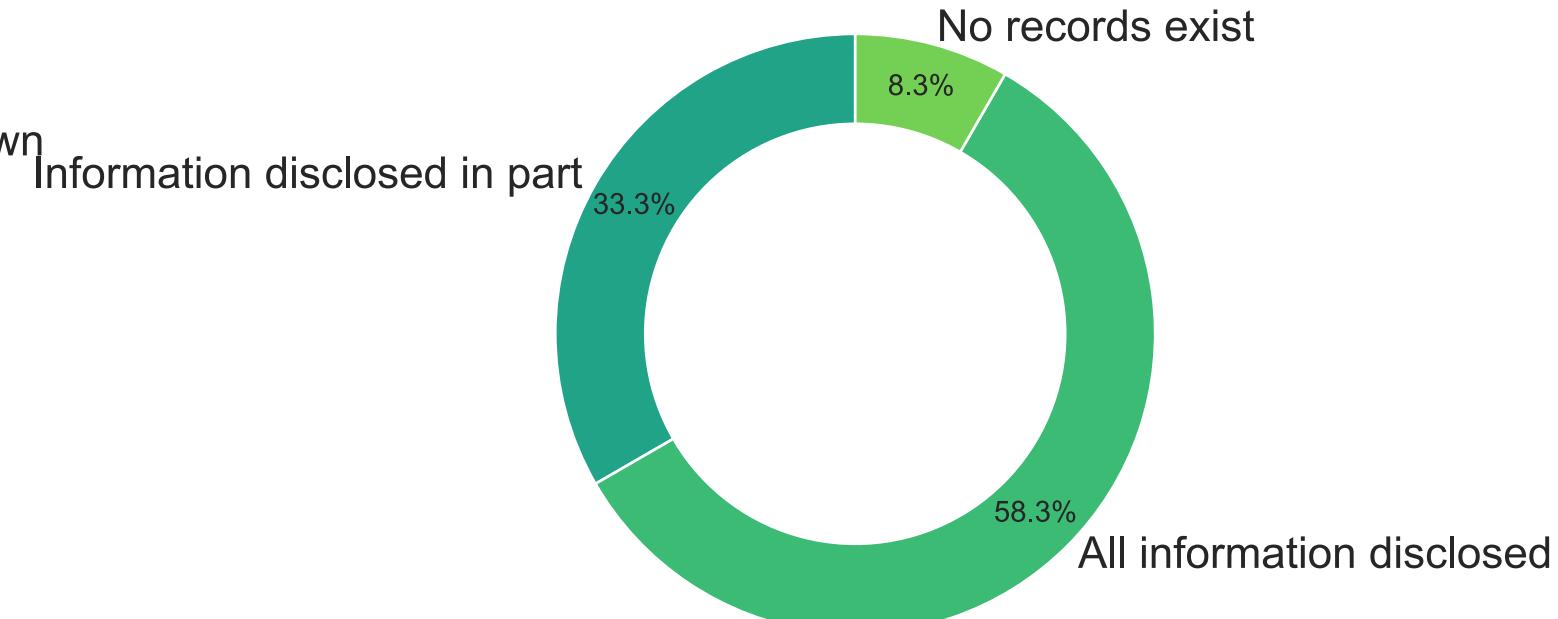


Decision percentage for each n-gram

grand river transit

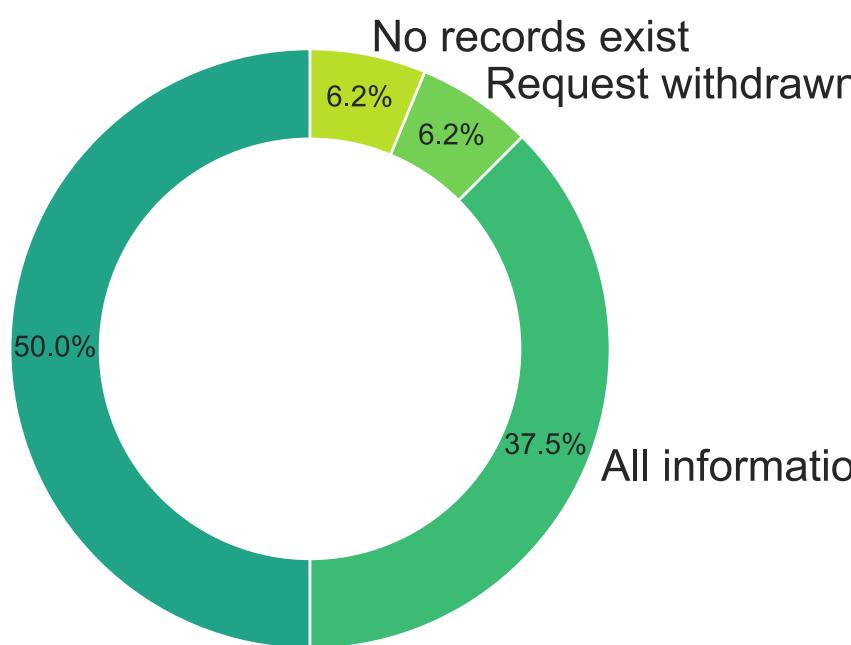


food bear illness



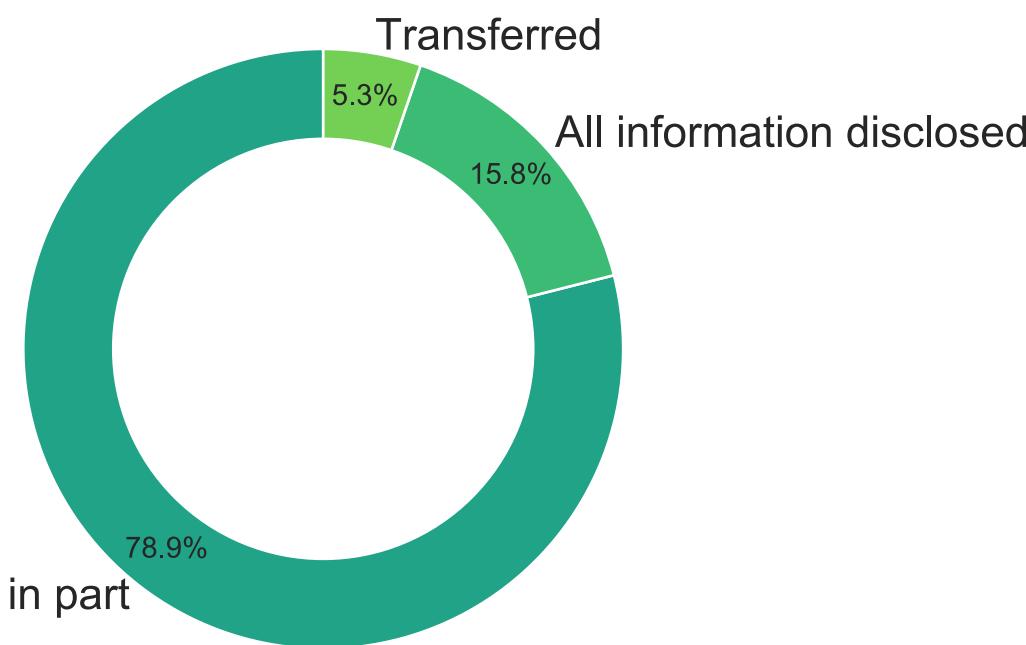
public health inspection

Information disclosed in part



rabies control

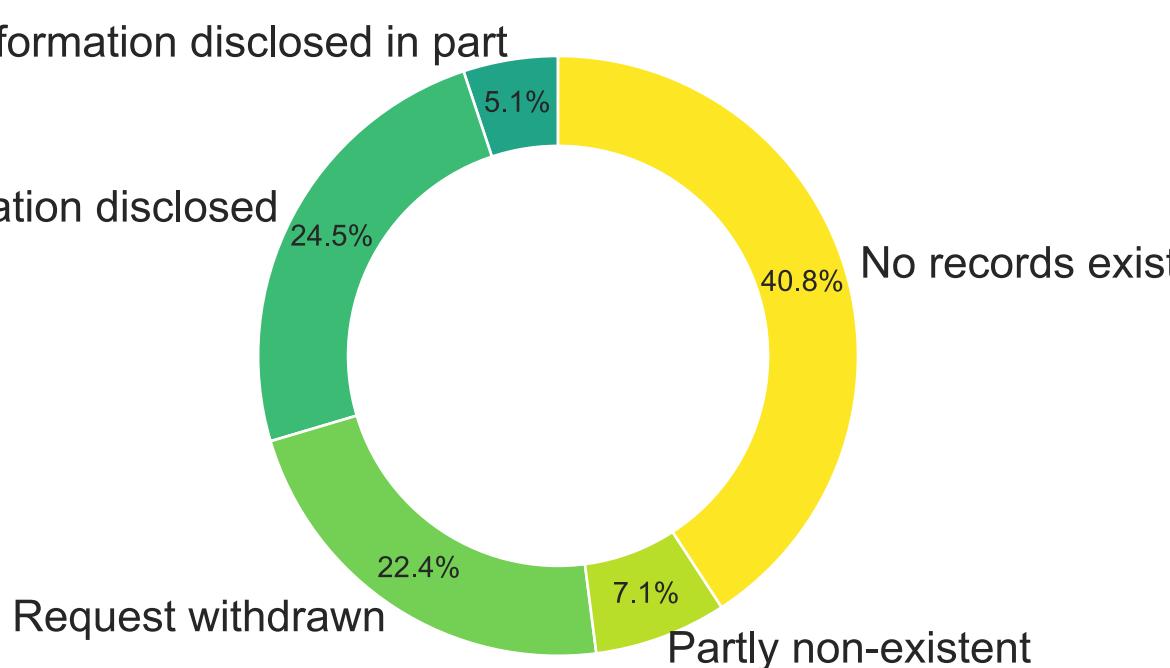
Information disclosed in part



environmental site

Information disclosed in part

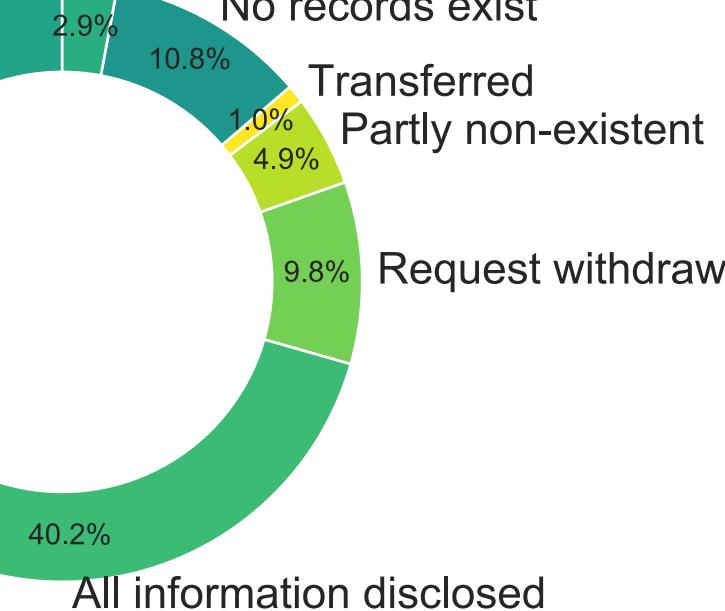
All information disclosed



Information disclosed in part

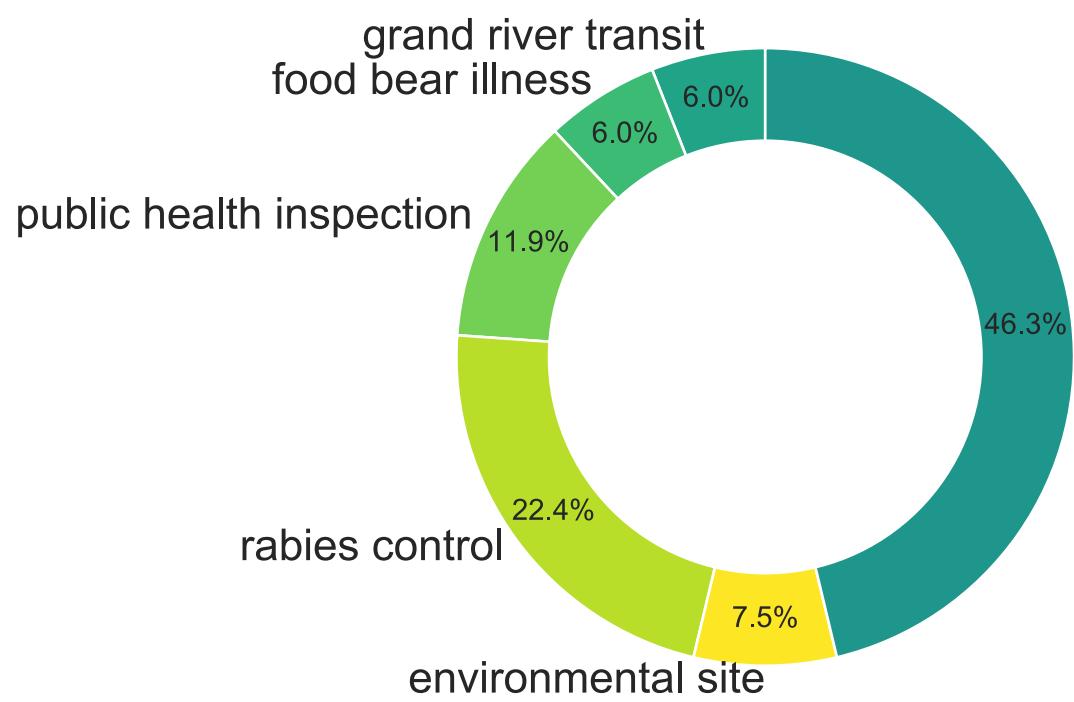
ontario works

Information disclosed in part

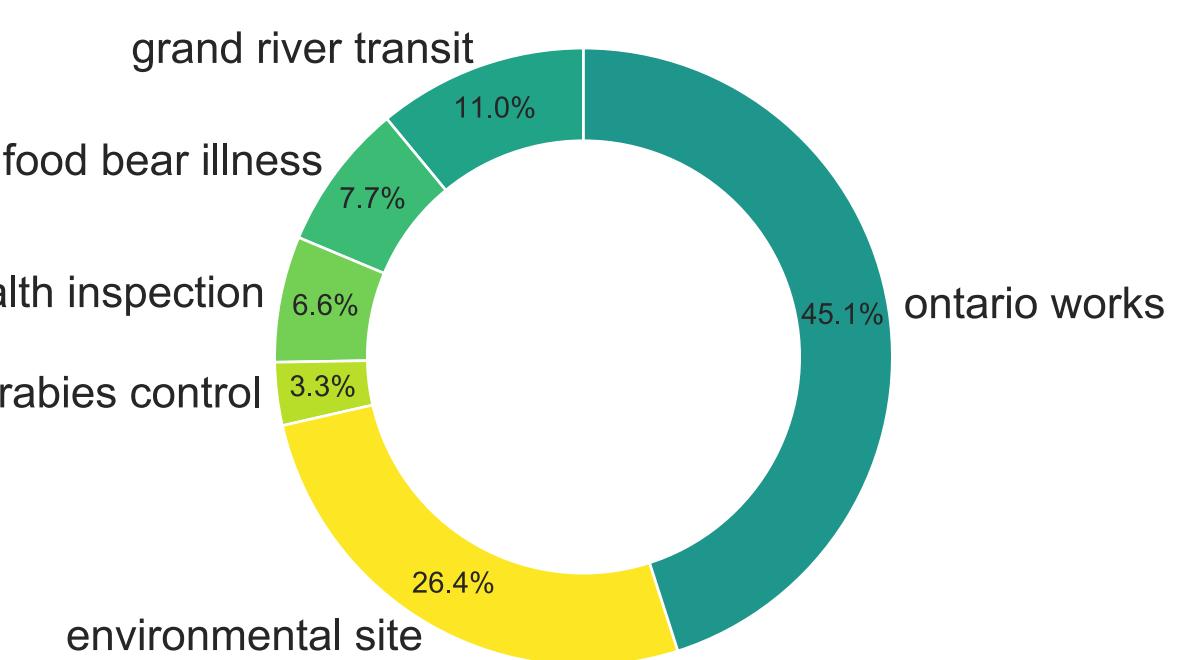


For requests with the n-grams, n-gram percentage based on decision

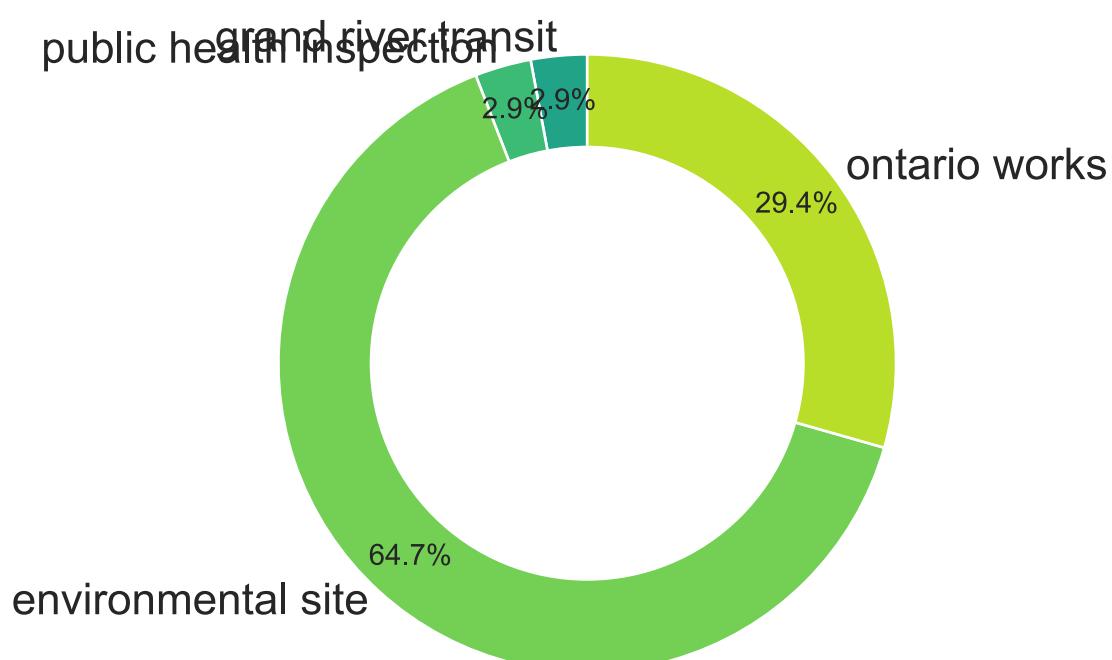
Information disclosed in part



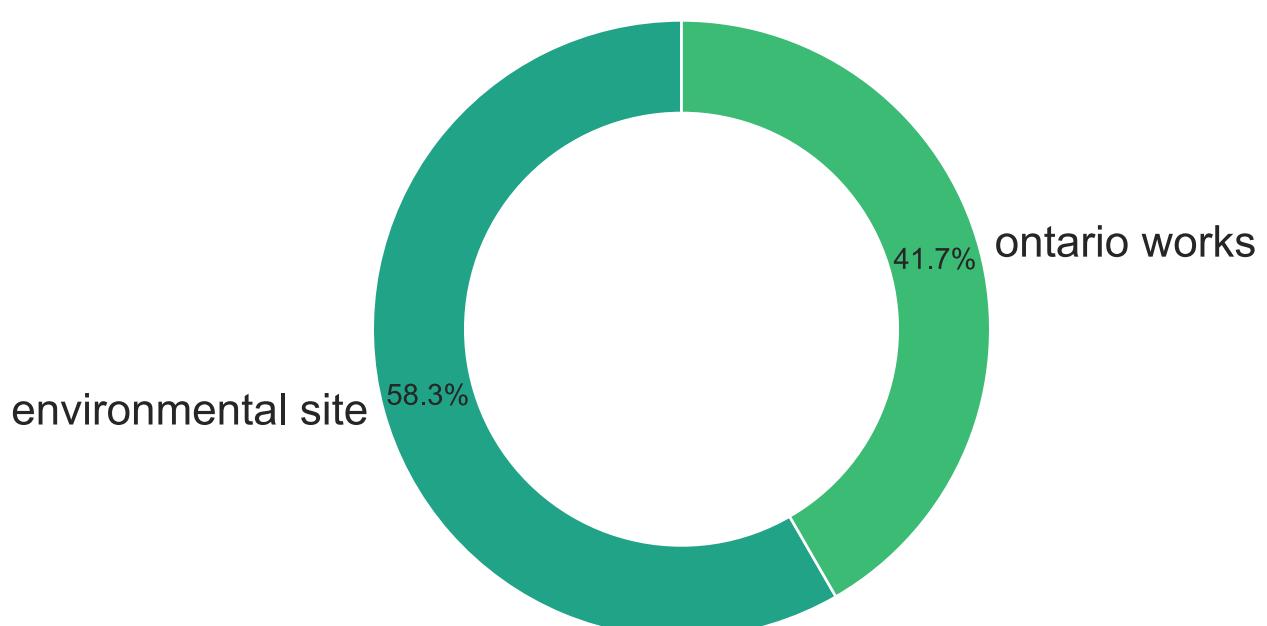
All information disclosed



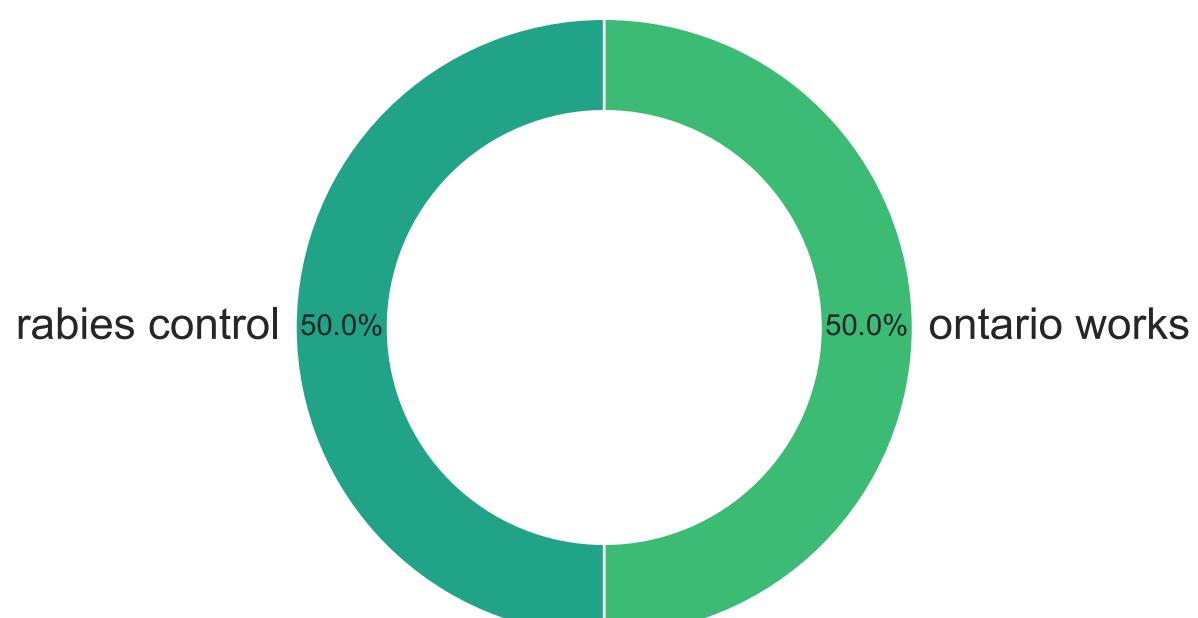
Request withdrawn



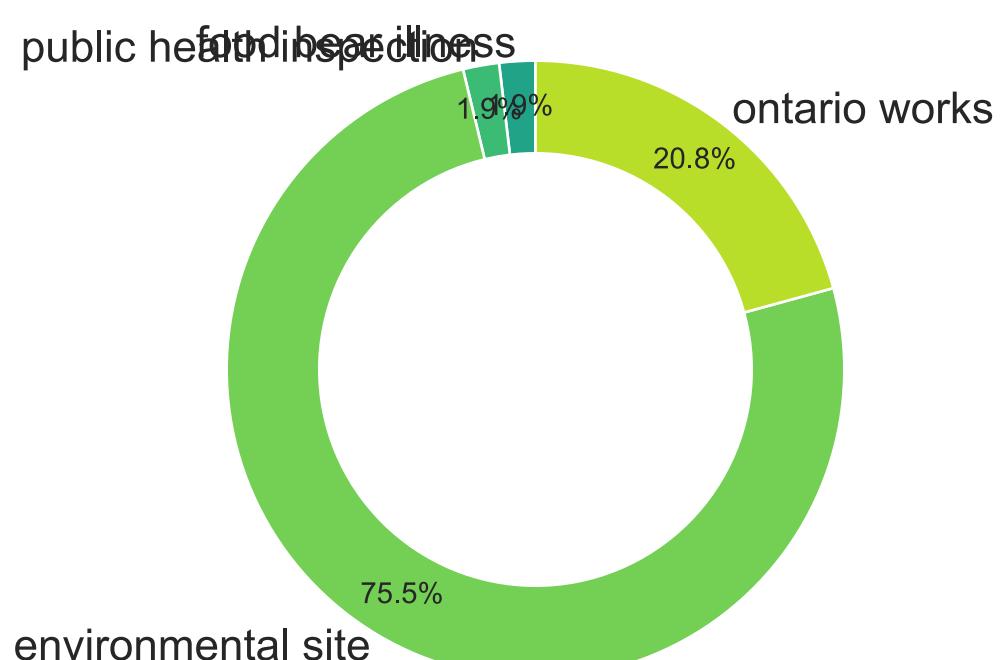
Partly non-existent



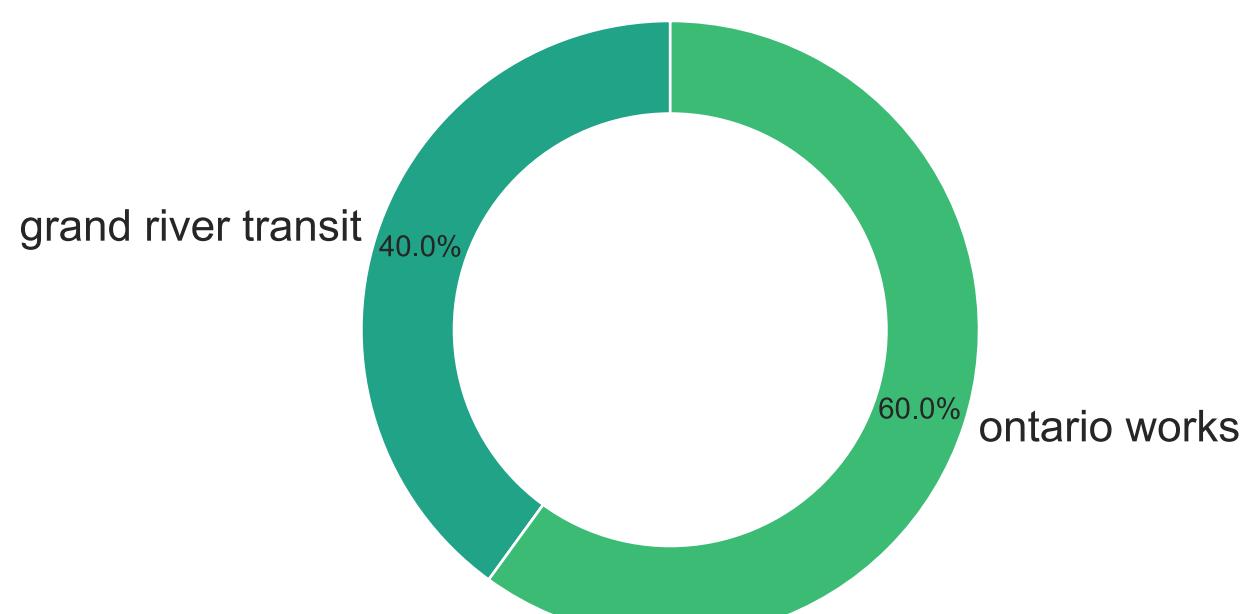
Transferred



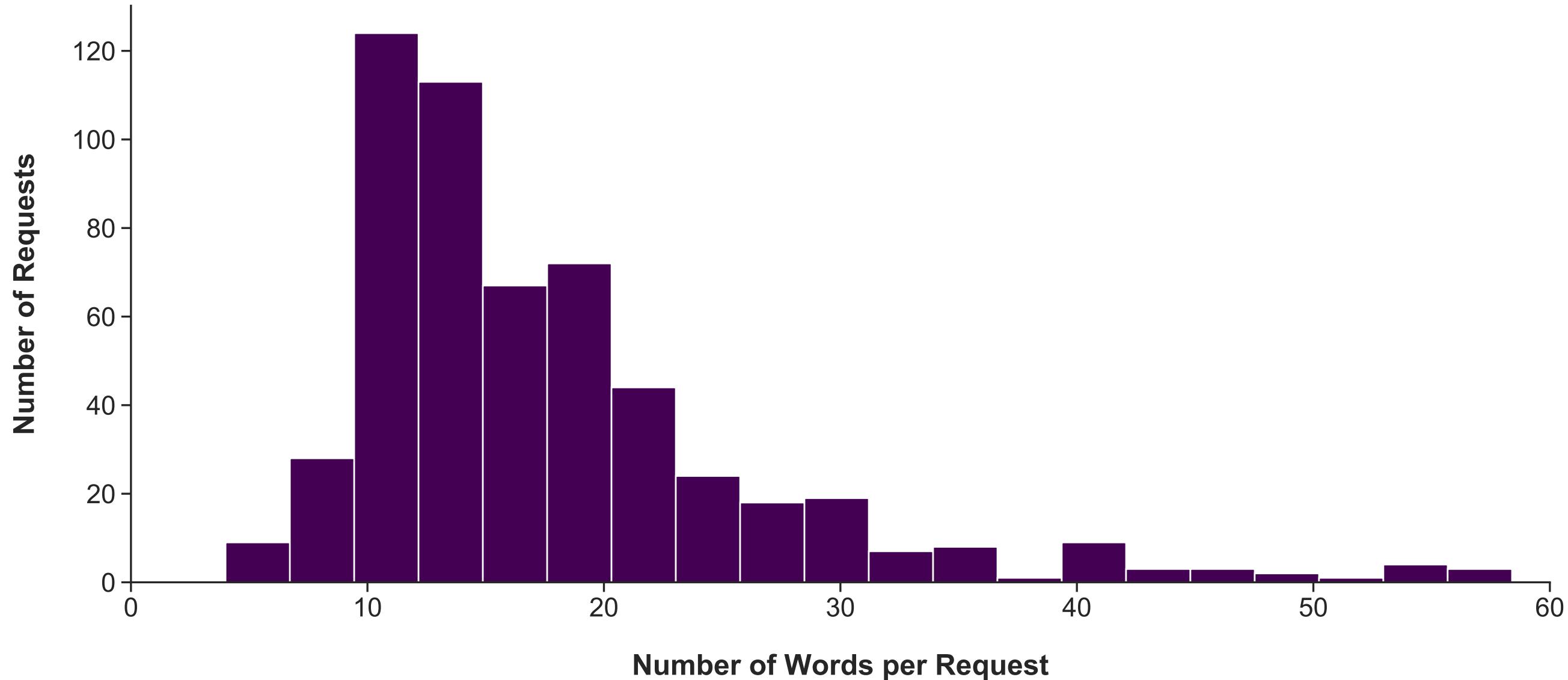
No records exist



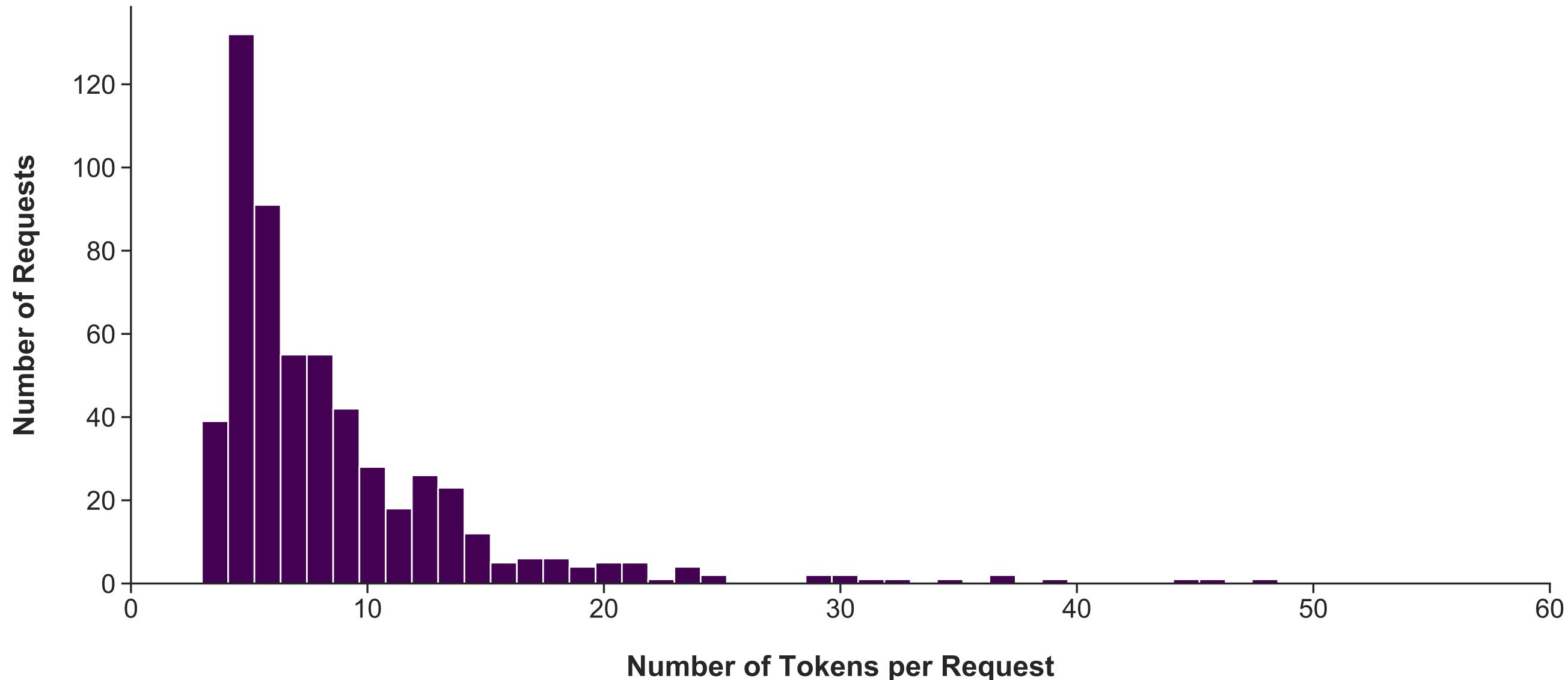
Abandoned



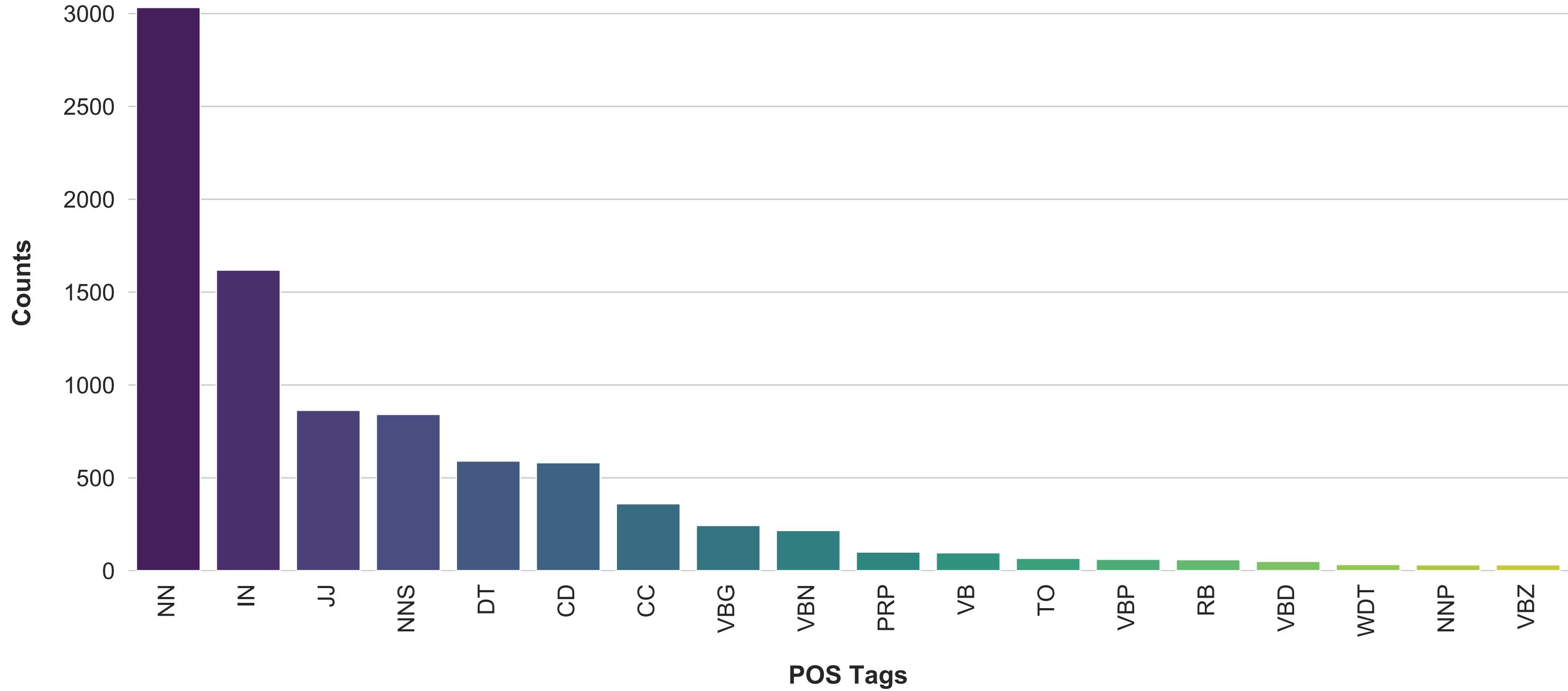
Average number of words per request is 20.5, while the median is 15.0



Average number of tokens per request is 9.3, while the median is 7.0

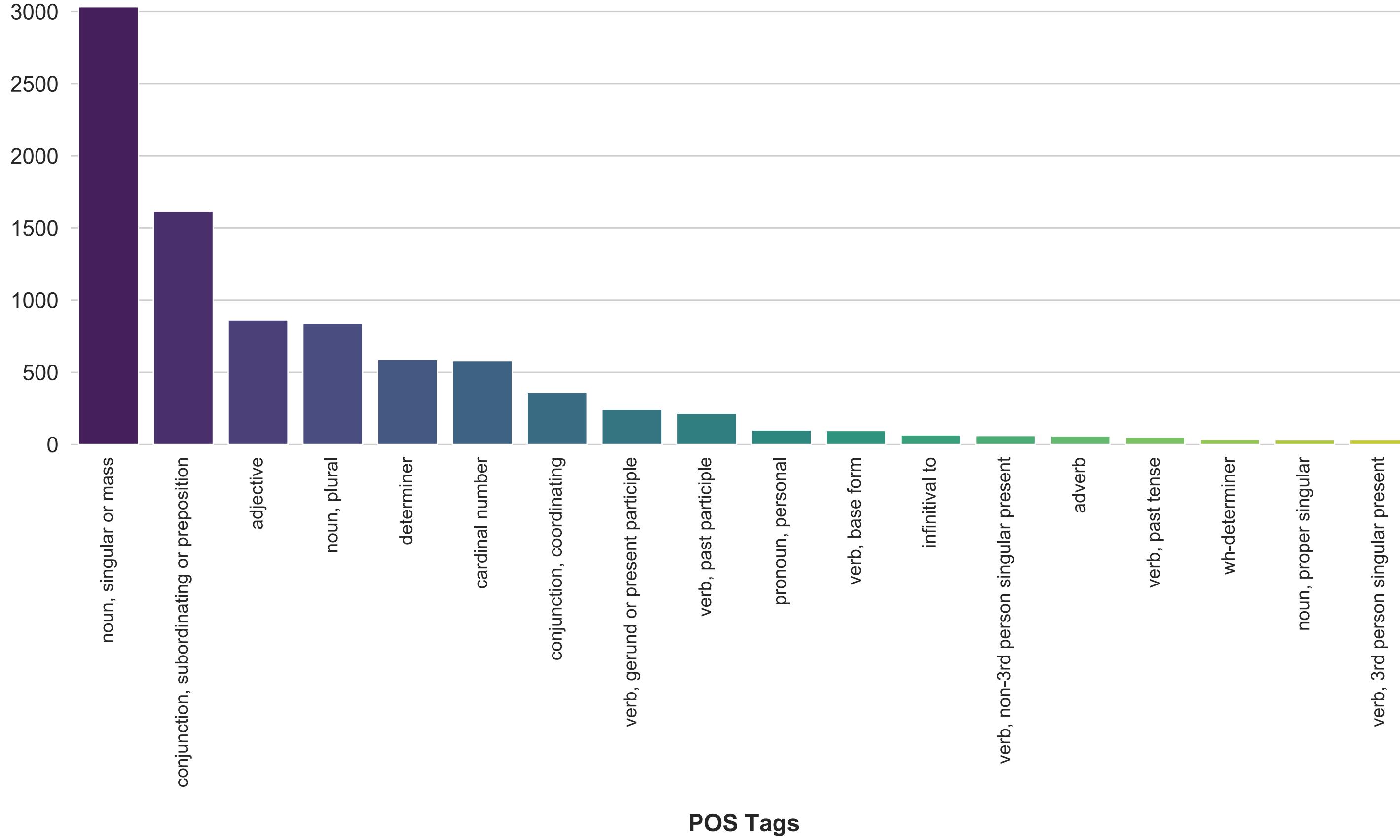


Full Tokenized Text

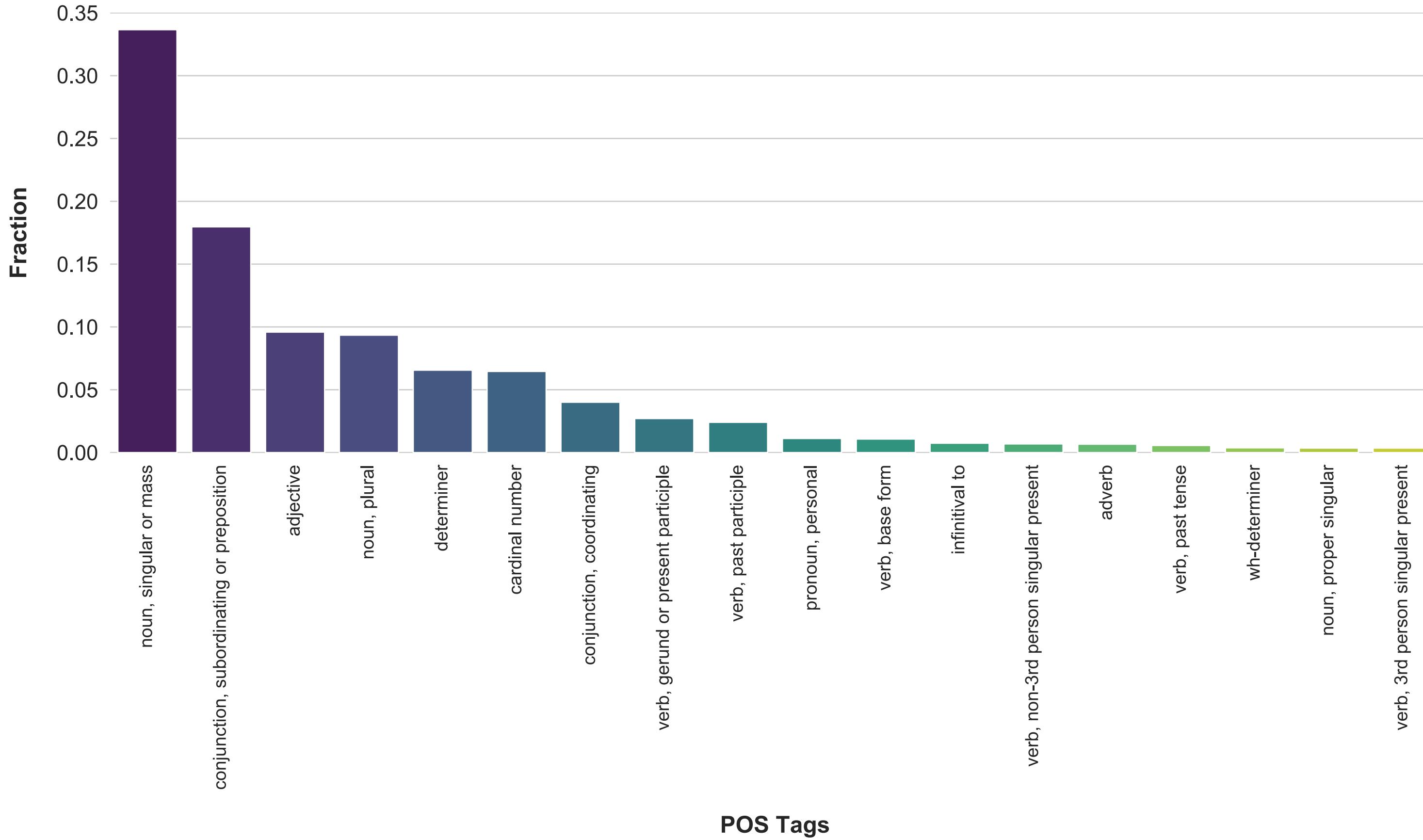


Full Tokenized Text

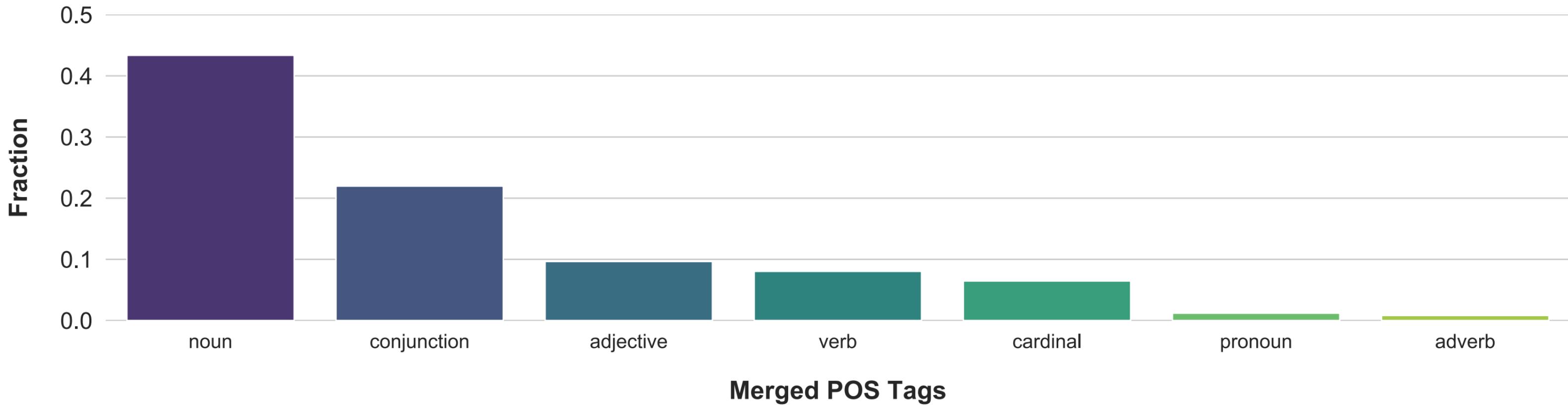
Counts



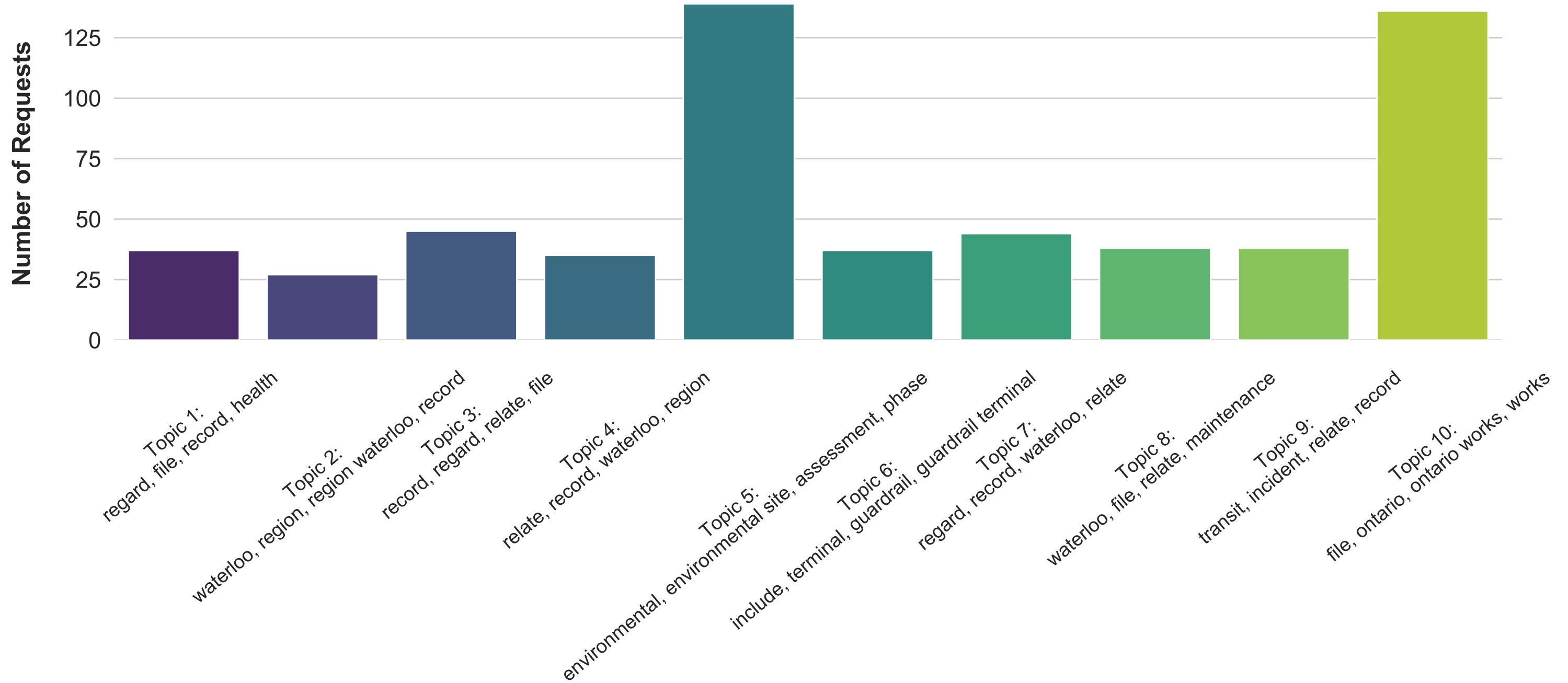
Full Tokenized Text



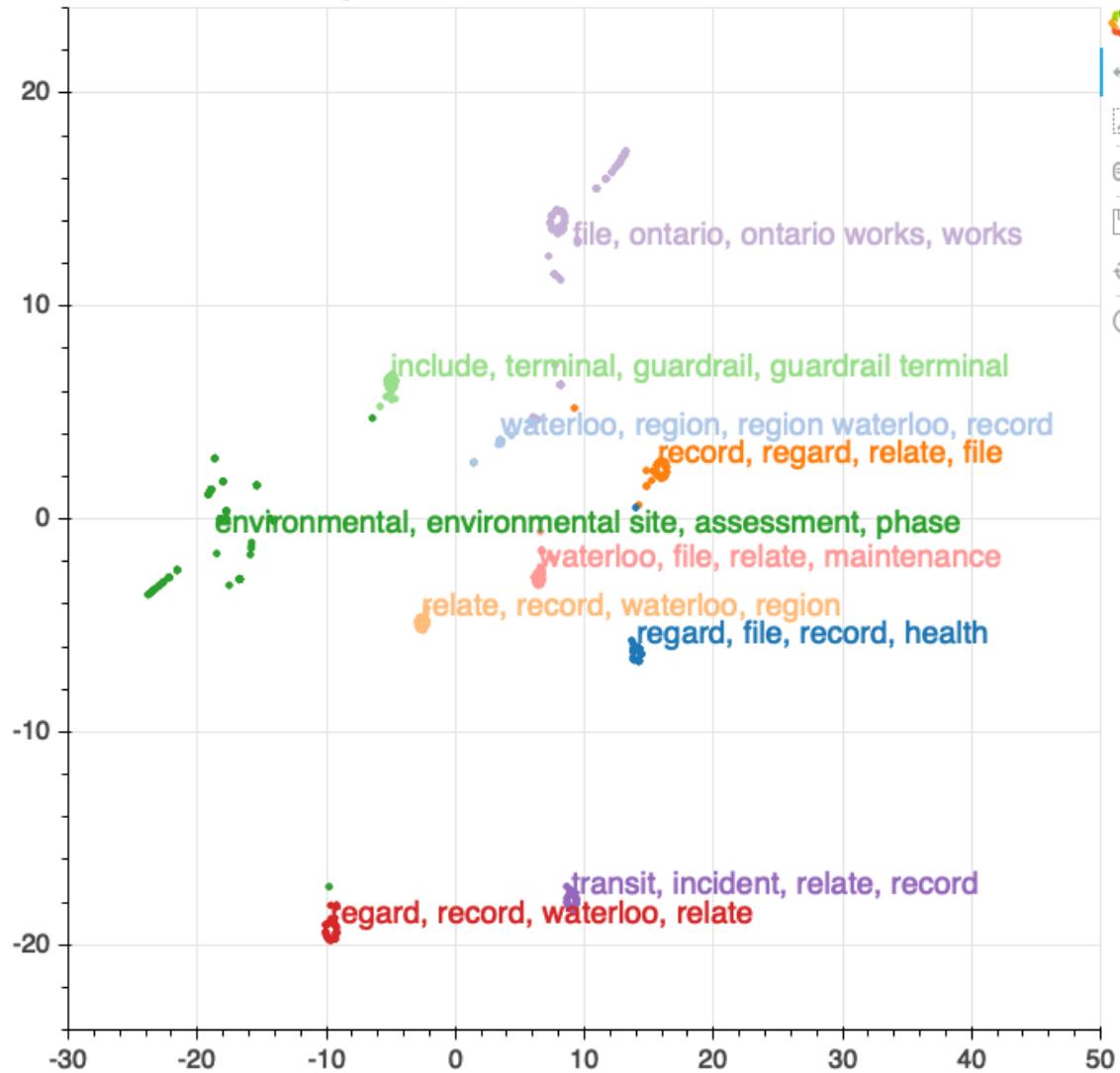
Full Tokenized Text



LDA Topic Counts - CountVectorizer

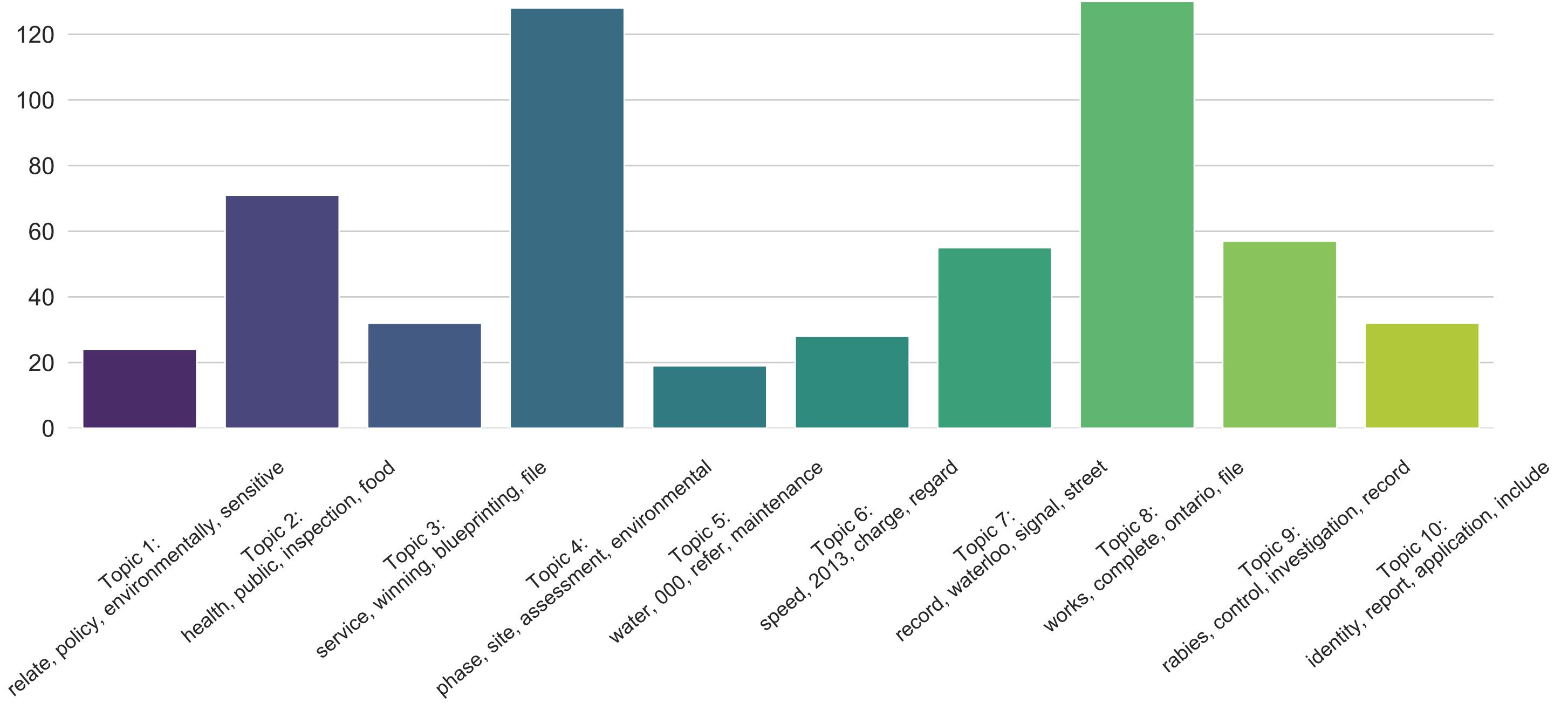


t-SNE Clustering of 10 LDA Topics - CountVectorizer

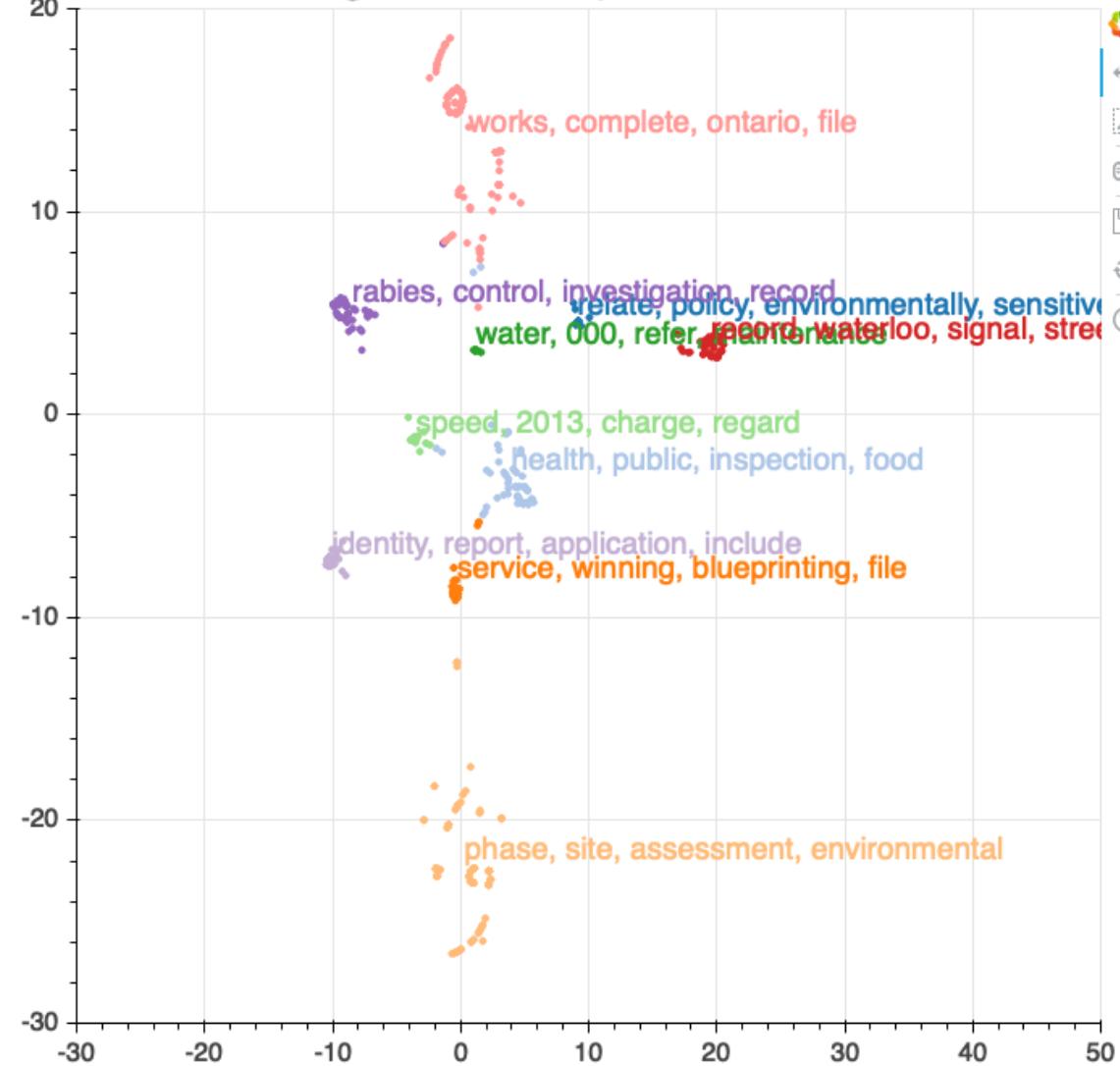


LDA Topic Counts - tf-idf Vectorizer

Number of Requests



t-SNE Clustering of 10 LDA Topics - tf-idf Vectorizer



LSA Topic Counts - CountVectorizer

Number of Requests

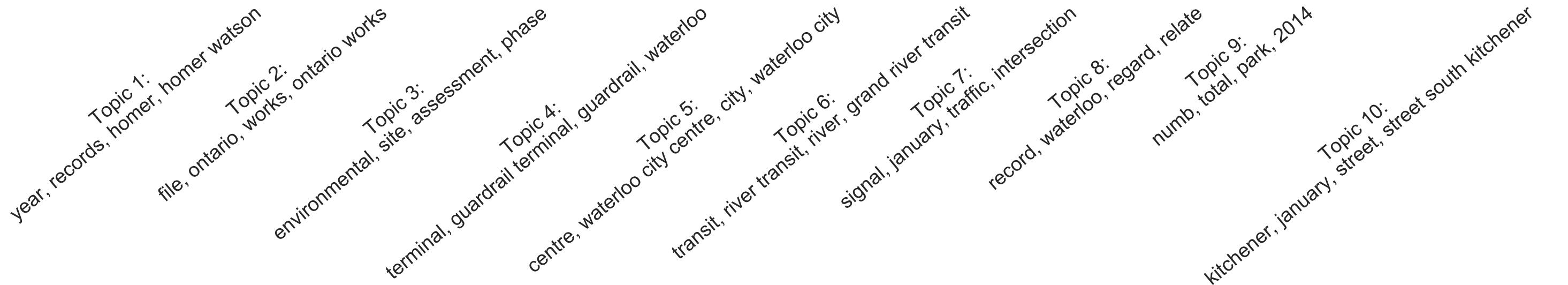
200

150

100

50

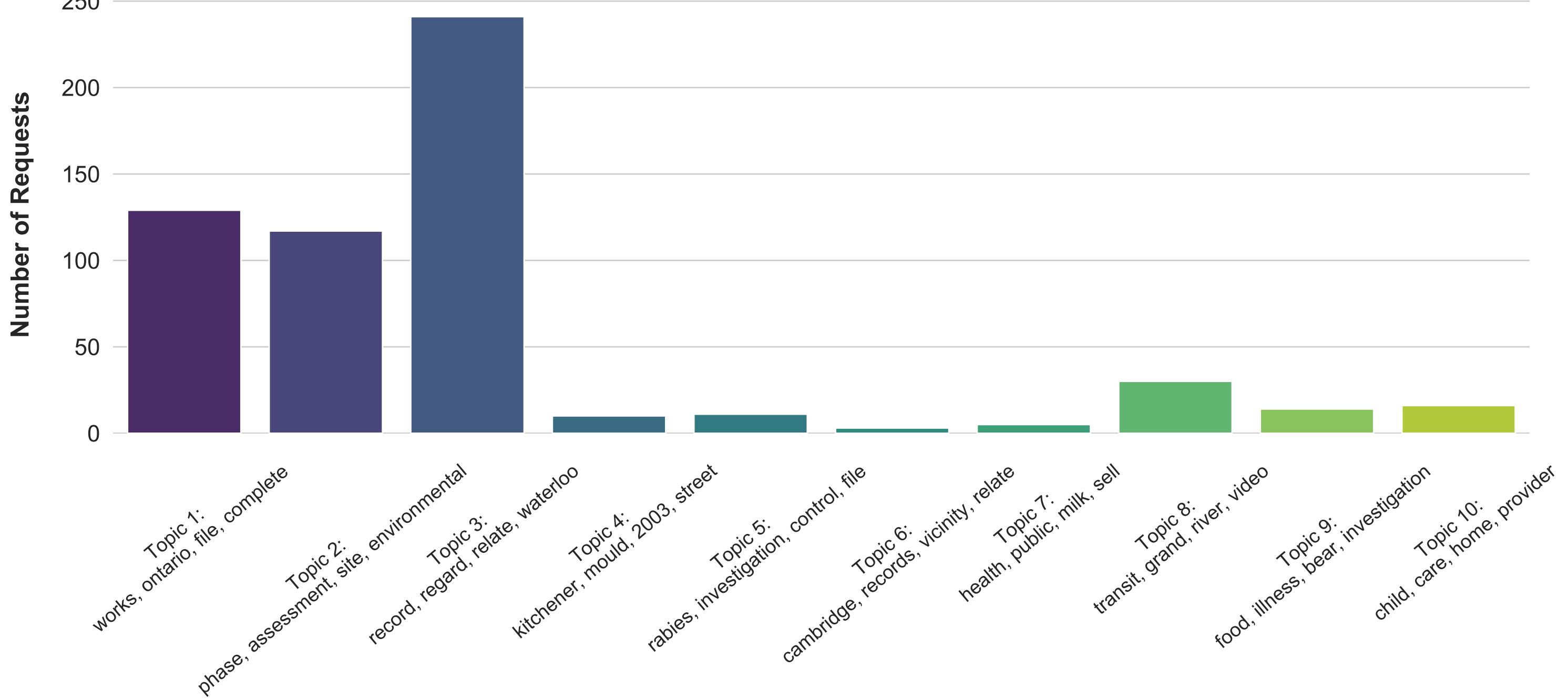
0



t-SNE Clustering of 10 LSA Topics - CountVectorizer



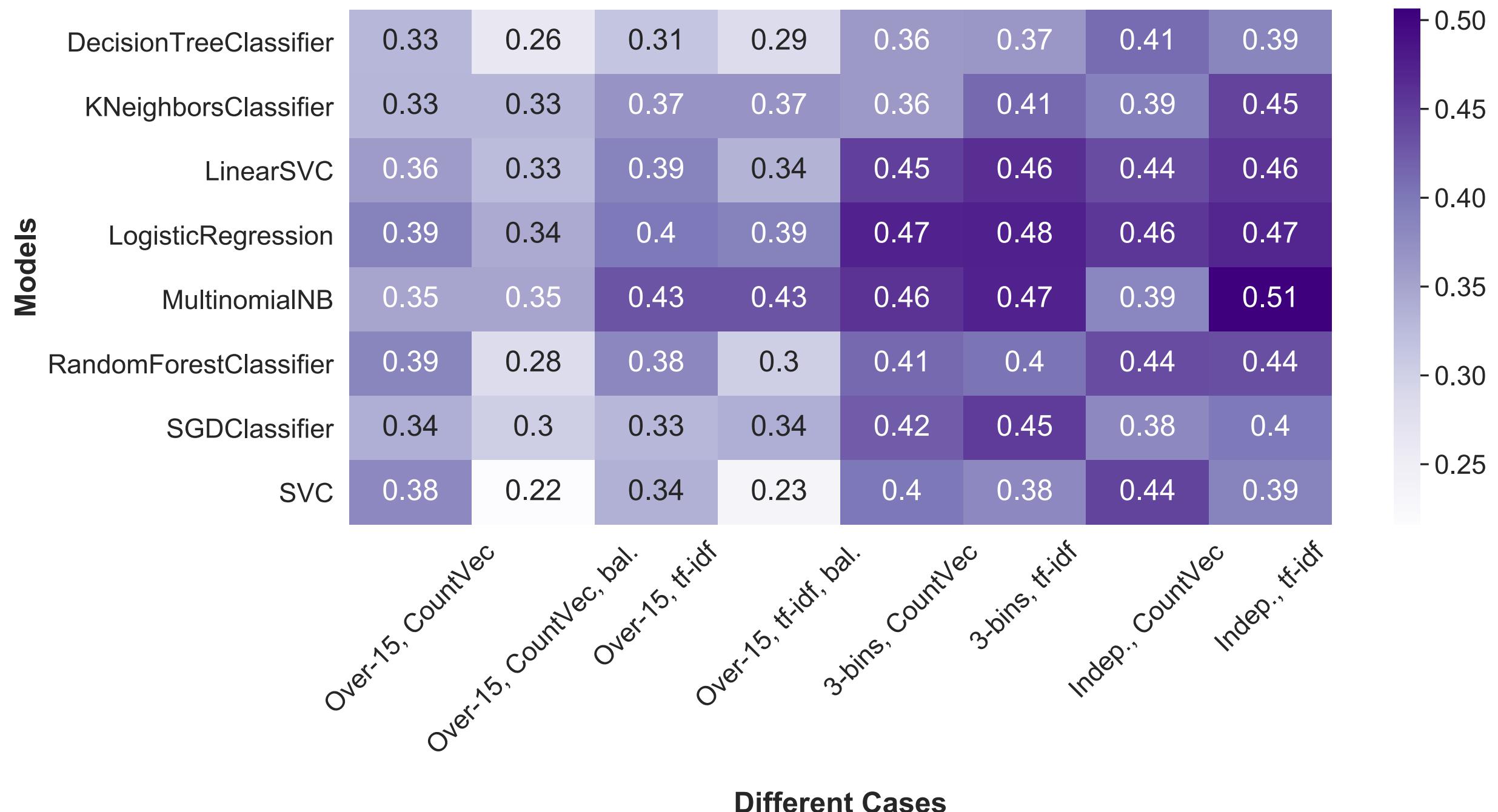
LSA Topic Counts - tf-idf Vectorizer



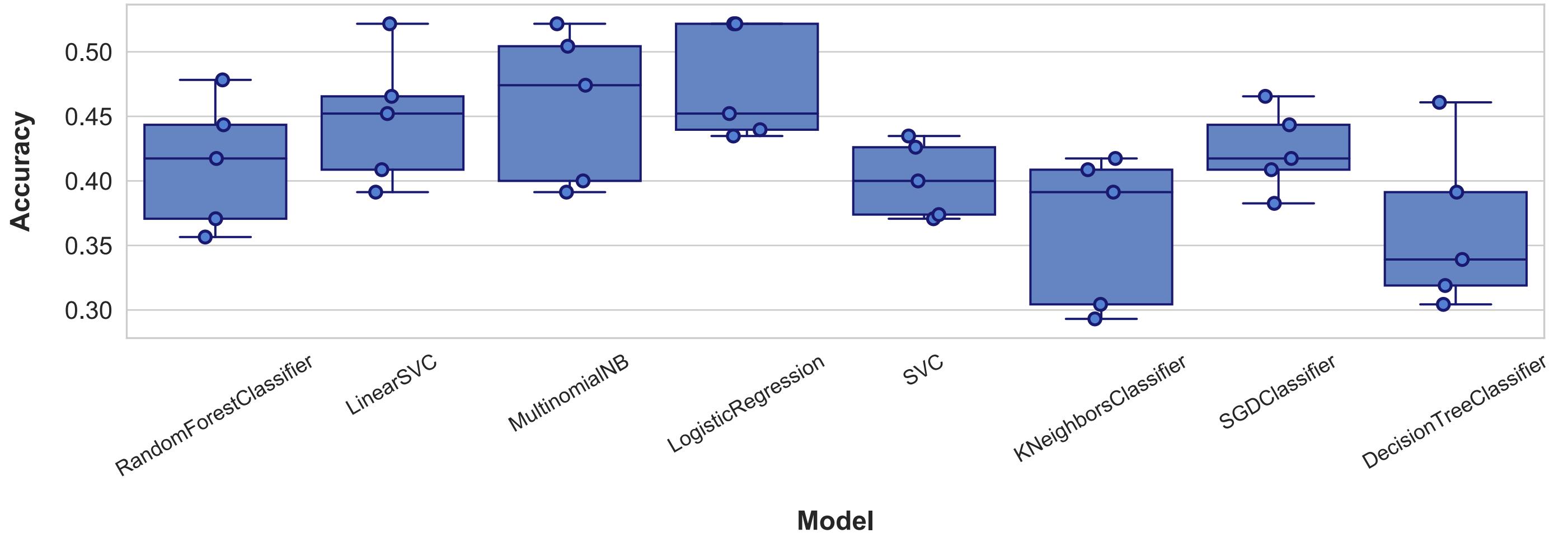
t-SNE Clustering of 10 LSA Topics - tf-idf Vectorizer



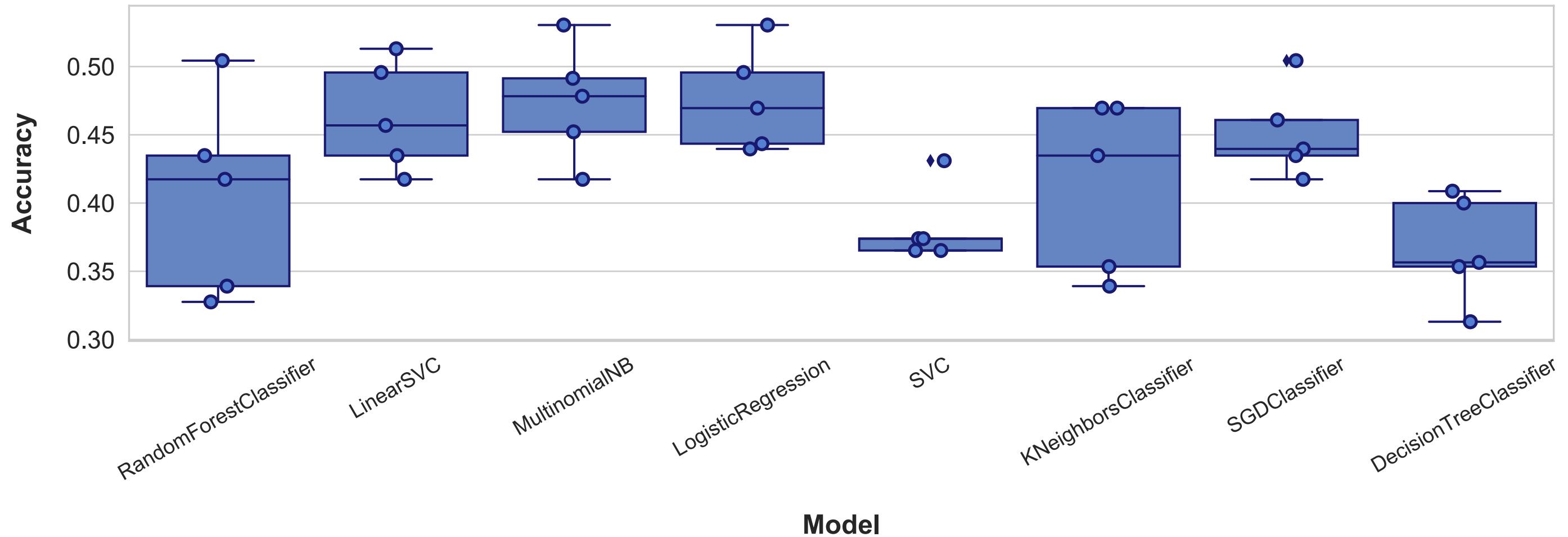
ML Model Accuracy



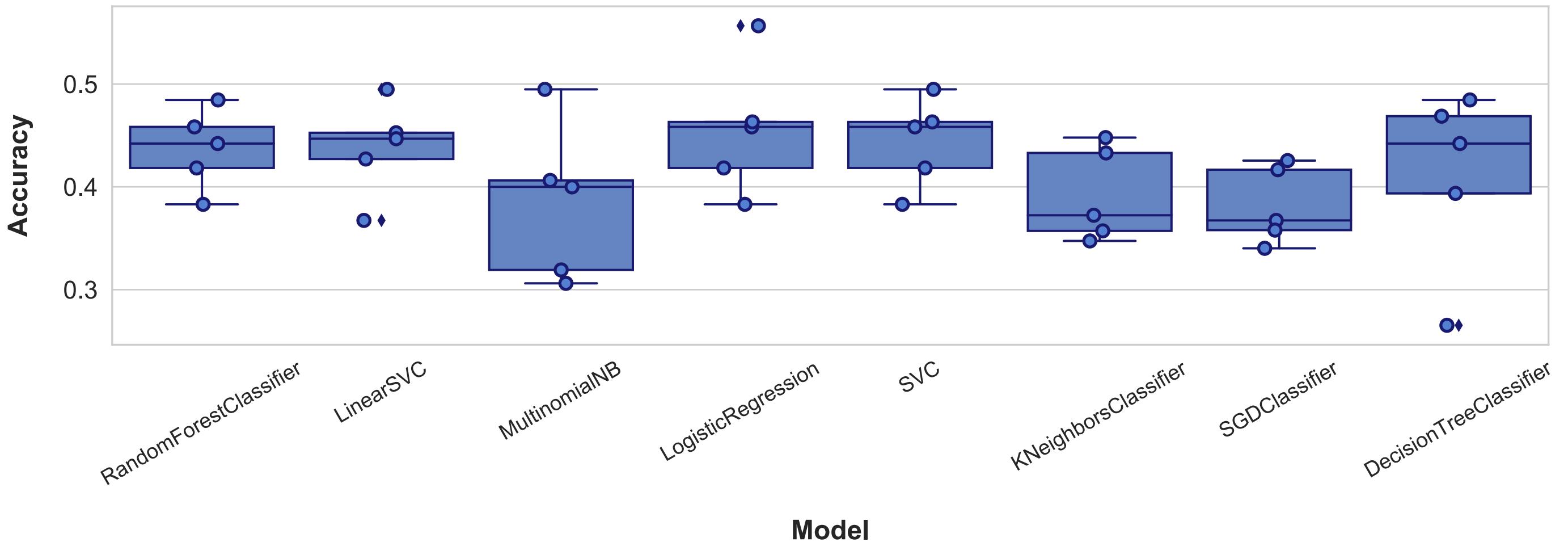
Classifier comparison for the 3-bin case, using CountVectorizer



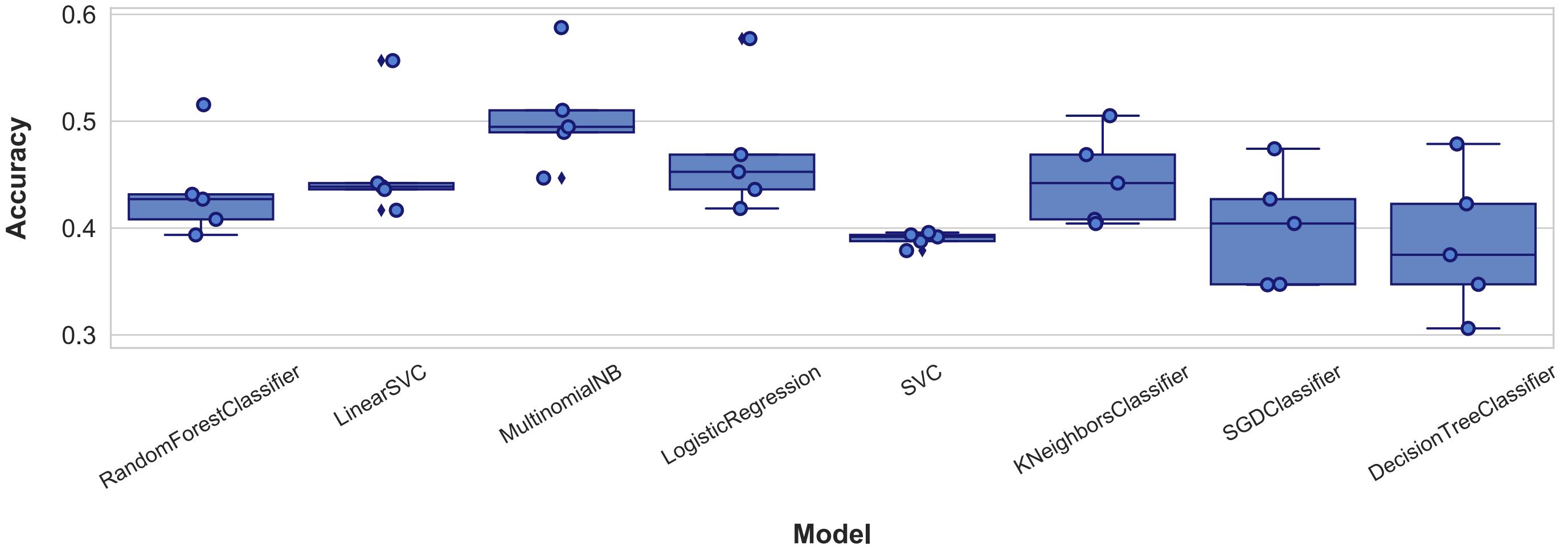
Classifier comparison for the 3-bin case, using tf-idf



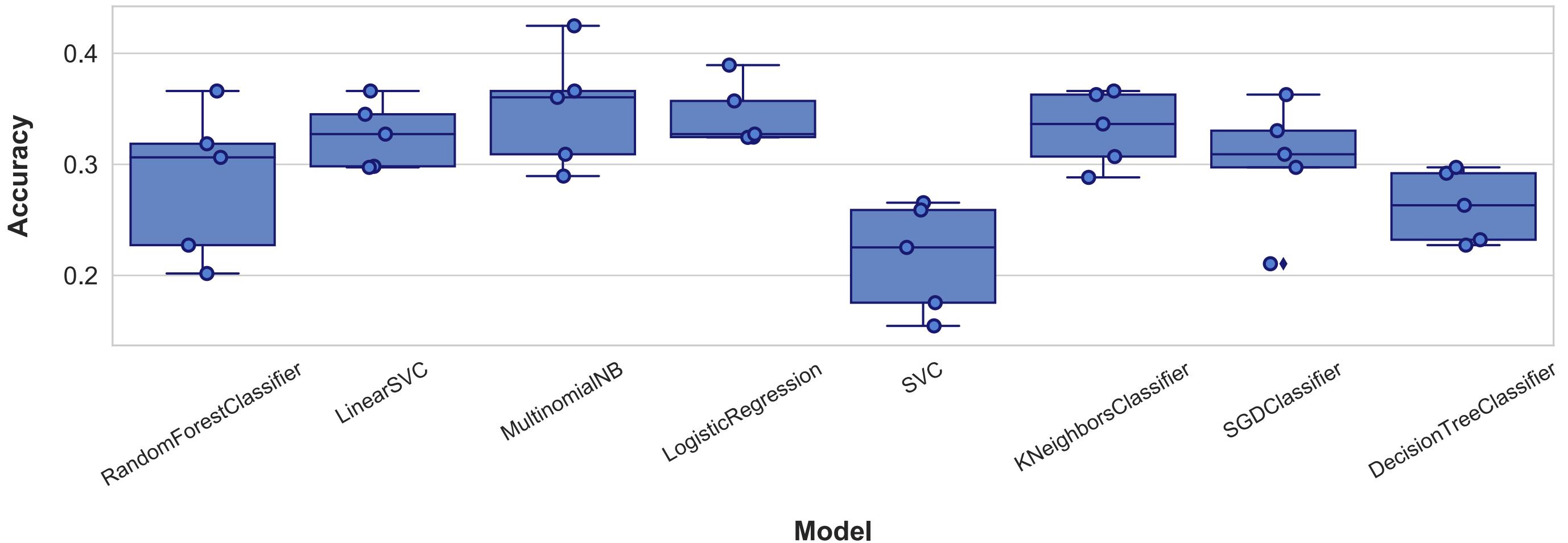
Classifier comparison for the indep. case, using CountVectorizer



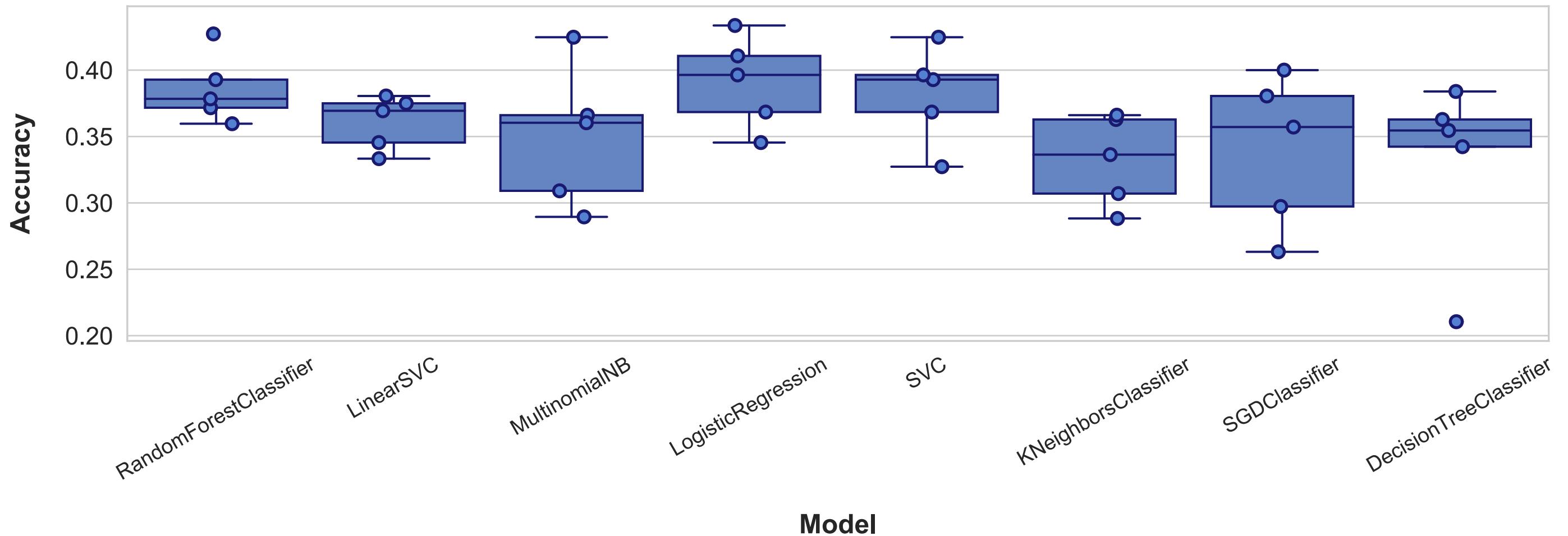
Classifier comparison for the indep. case, using tf-idf



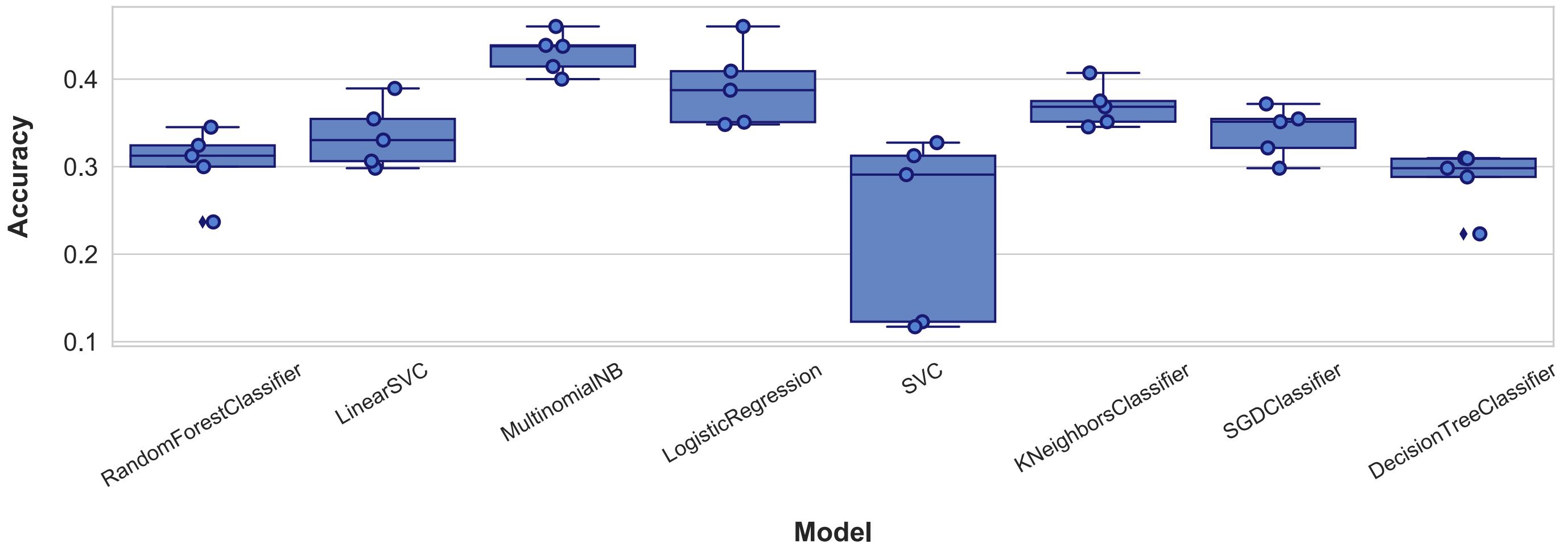
Classifier comparison for the over 15 case, using CountVectorizer, Balanced



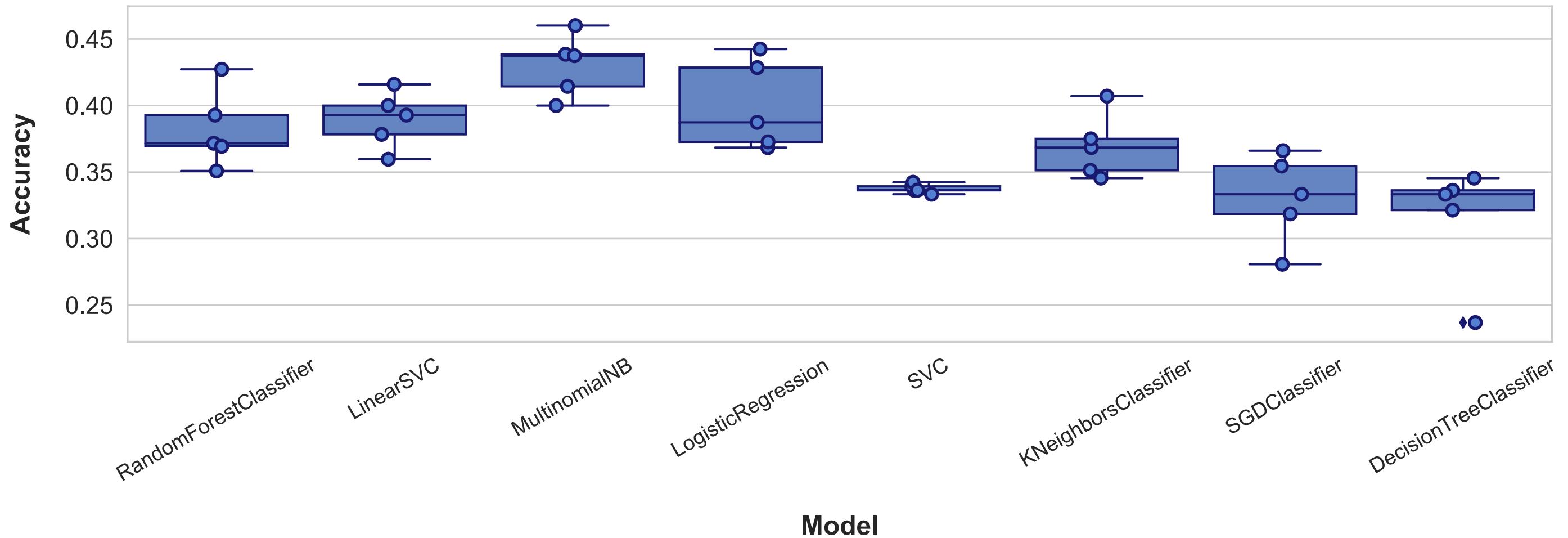
Classifier comparison for the over 15 case, using CountVectorizer



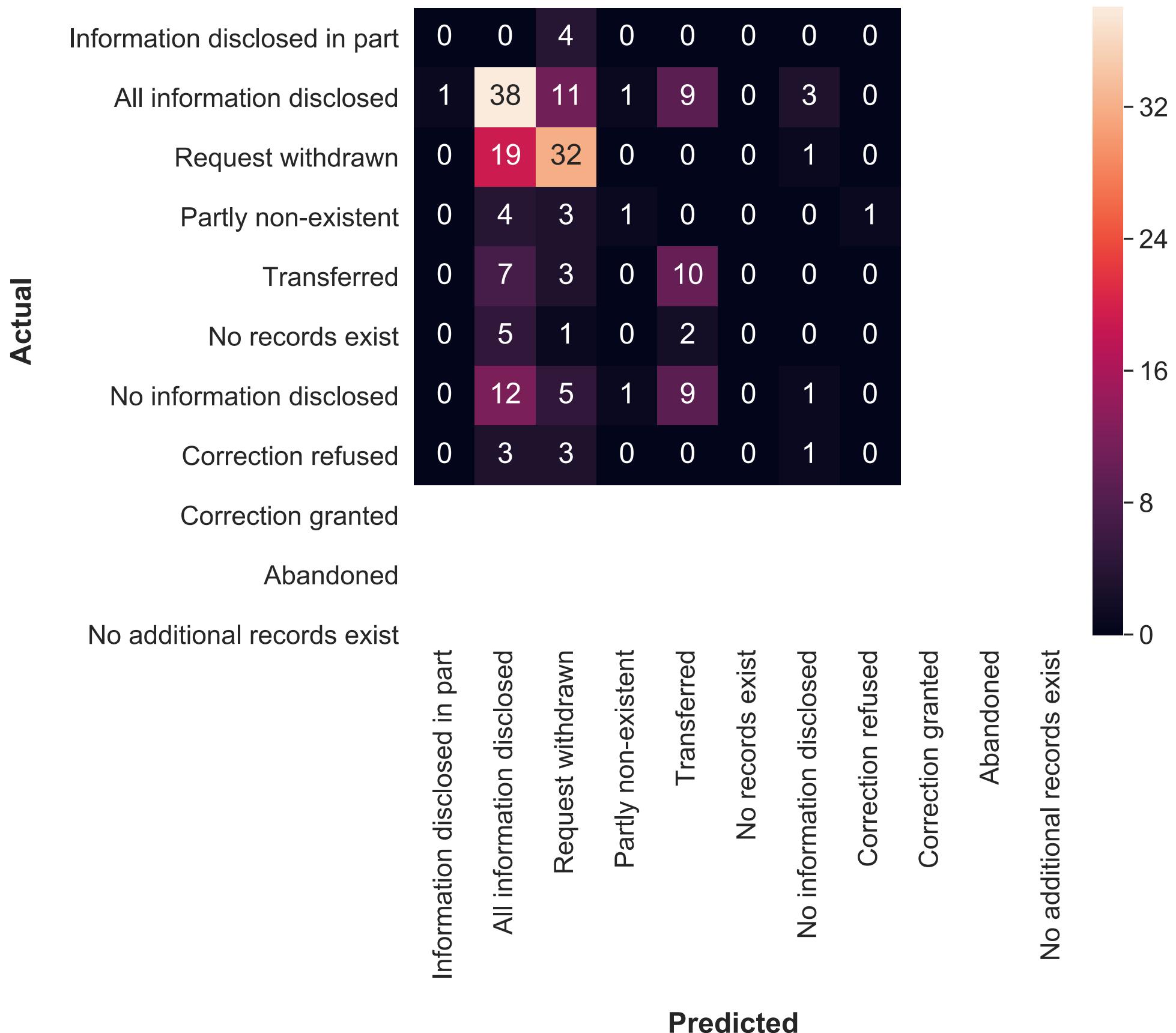
Classifier comparison for the over 15 case, using tf-idf, balanced



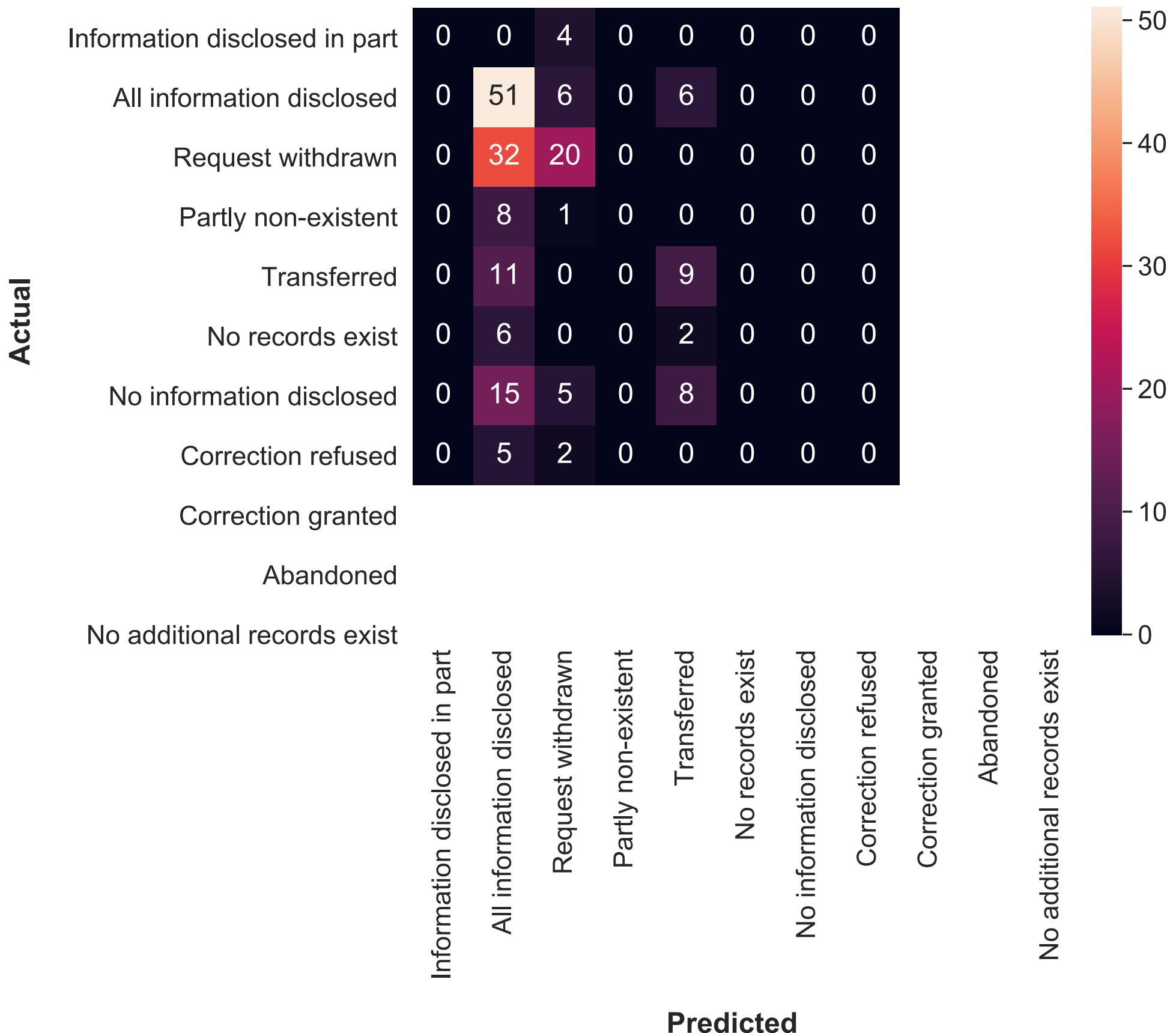
Classifier comparison for the over 15 case, using tf-idf



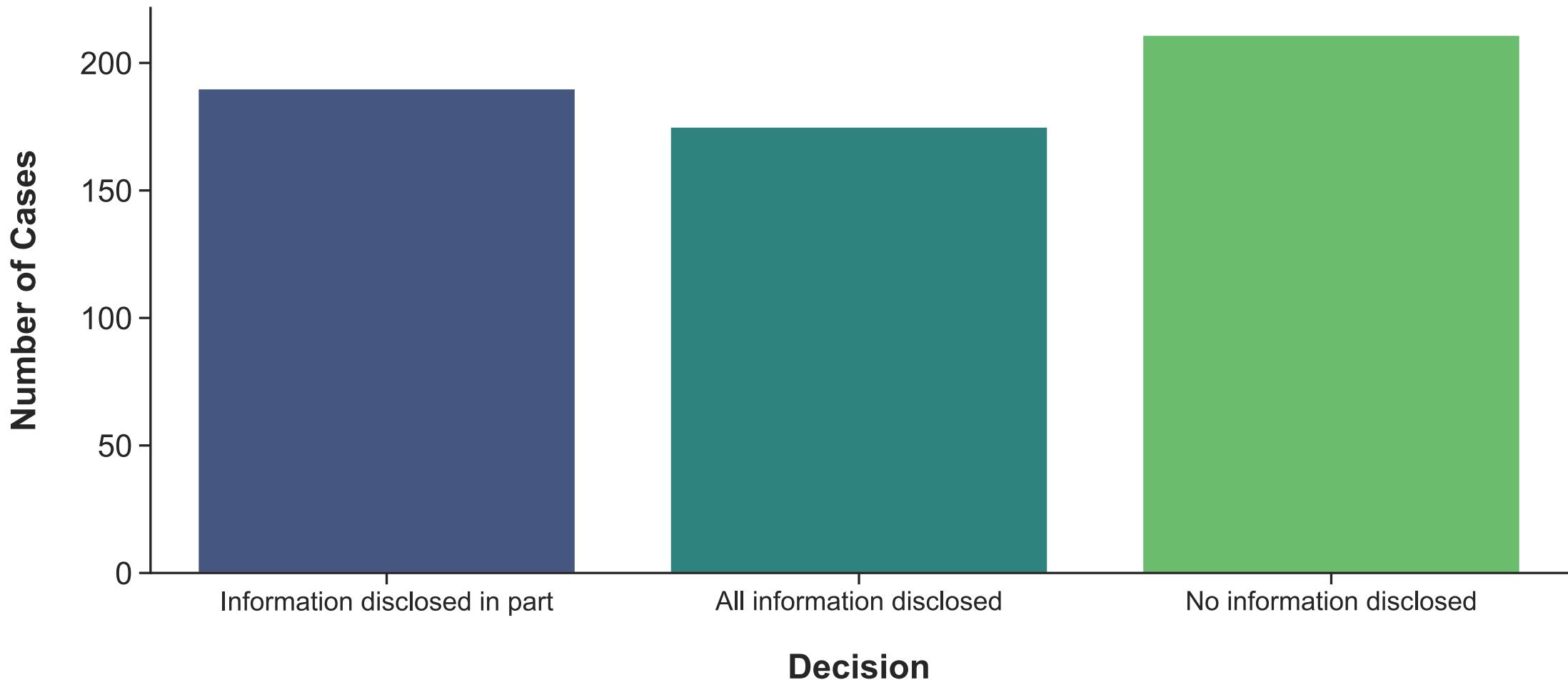
MultinomialNB, CountVectorizer, full set



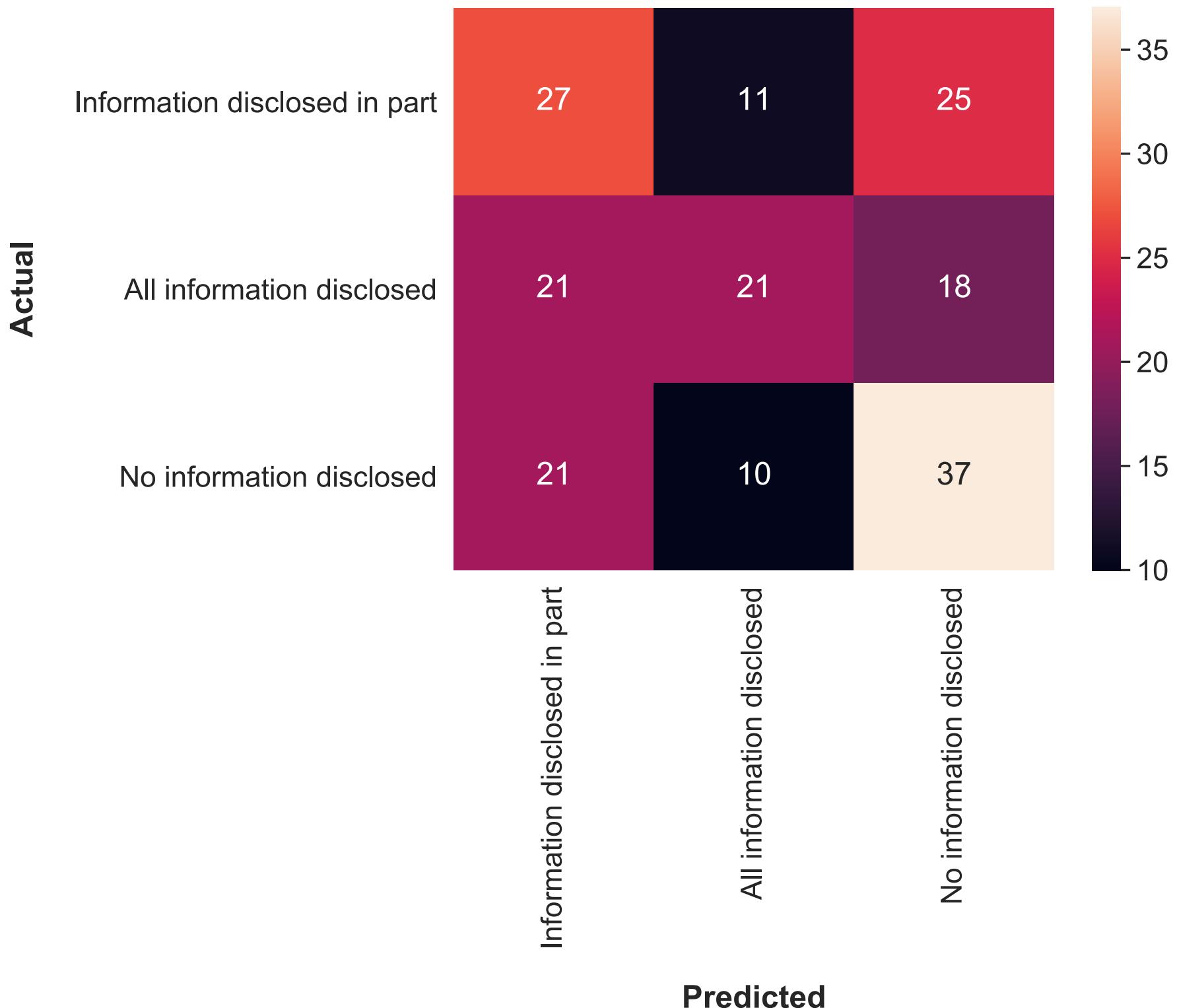
MultinomialNB, tf-idf, full set



Full data split into three categories only

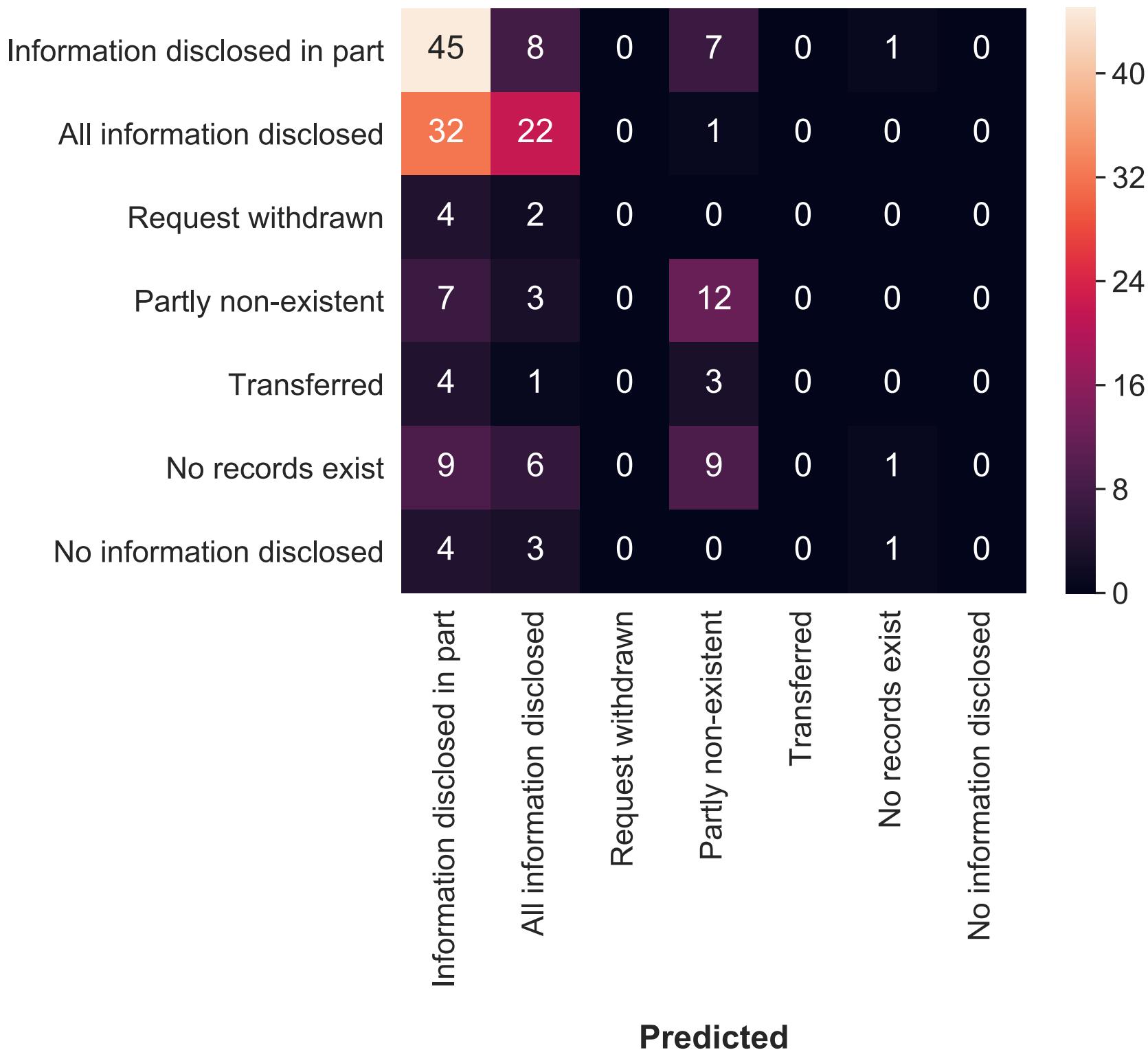


LogisticRegression, tf-idf, 3 bins



MultinomialNB, tf-idf, over 15

Actual



MultinomialNB, tf-idf, indep.

