FELADATKIÍRÁS

A feladatkiírást a **tanszék saját előírása szerint** vagy a tanszéki adminisztrációban lehet átvenni, és a tanszéki pecséttel ellátott, a tanszékvezető által aláírt lapot kell belefűzni a leadott munkába, vagy a tanszékvezető által elektronikusan jóváhagyott feladatkiírást kell a Diplomaterv Portálról letölteni és a leadott munkába belefűzni (ezen oldal HELYETT, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell megismételni a feladatkiírást.



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

XXX Tanszék

Péter Bircher

Analysis of legal documents using text mining techniques

Supervisor

Csaba Gáspár

BUDAPEST, 2022

Tartalomjegyzék

[Összefoglaló 6](#_Toc59896130)

[Abstract 7](#_Toc59896131)

[1 Bevezetés 8](#_Toc59896132)

[1.1 Formázási tudnivalók 8](#_Toc59896133)

[1.1.1 Címsorok 8](#_Toc59896134)

[1.1.2 Képek 8](#_Toc59896135)

[1.1.3 Kódrészletek 8](#_Toc59896136)

[1.1.4 Irodalomjegyzék 8](#_Toc59896137)

[2 Utolsó simítások 10](#_Toc59896138)

[Irodalomjegyzék 11](#_Toc59896139)

[Függelék 12](#_Toc59896140)

Hallgatói nyilatkozat

Alulírott **Rezeda Kázmér**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot/ diplomatervet (nem kívánt törlendő) meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2022. 04. 22.

...…………………………………………….

Rezeda Kázmér

Összefoglaló

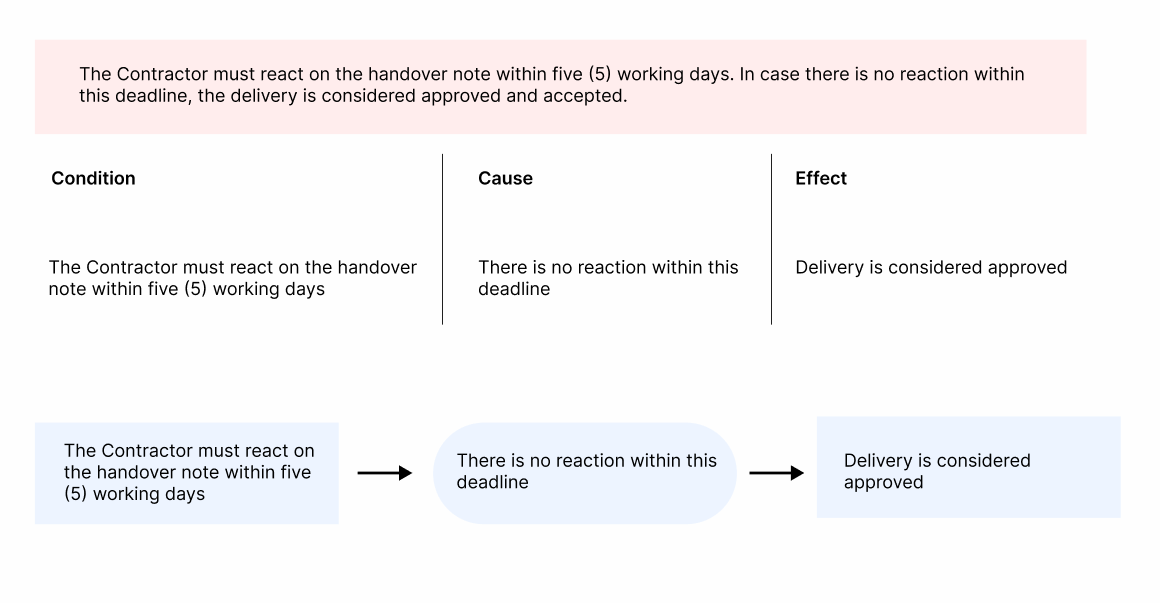
Ide jön a ½-1 oldalas magyar nyelvű összefoglaló, melynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

Abstract

Ide jön a ½-1 oldalas angol nyelvű összefoglaló, amelynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

# Logical expression extraction

Conditionals describe the result of a certain condition. These sentences are statements of an “if-then”, “unless-then” situation (although “then is not used”), but other keyword such as “when”, “where” or “in cases” are can be found in these kinds of sentences either. In legal documents, especially contracts, often contain parts where certain paragraphs come into force due to the fulfillment or non-fulfillment of a condition. What if there is a tool that is able to identify these structures in a document and could create a diagram next to document to help the readers better and quicker understanding the particular paragraph. In this study, we conduct an experiment in which we try to collect the conditional sentences found in the legal documents, then break them down into cause and effect and display them in a flowchart-like manner.

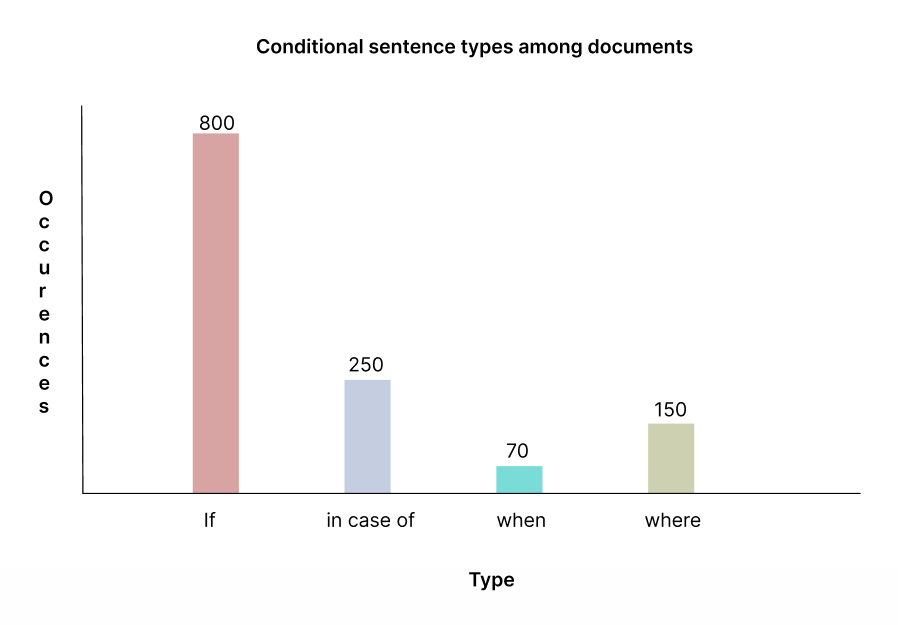


## Nature of the sentences

The word “if” can almost always be used to introduce a case or condition in legal documents. However, there are cases, when other words are more appropriate choice, than a simple “if-else” structure, it depends heavily on the context and what we want to express. In case of time or timing is important to the rule or describing a rare or once-only situation, or there is some certainty that an event will occur, “when” is a better choice. “Where” usually is to introduce adverbial clauses that refer to a specific physical place. Of course, this does not preclude the use of “in cases where”, “where”, or “in circumstances where” when they seem more logical and natural in the particular context.

The dataset, which I used in my following experiments, consists of 10 legal contracts. As previously mentioned, they come in a plain unformatted .txt format. In these documents, we can find examples of all above mentioned keywords. Examining the distribution of the conditional sentences among the documents, we find a very similar amount of each type.

Sentences with “if” ~ 800 occurrences, “in case of” or similar is around 200-250 occurrences. The “when” keyword usually listed around 70 occurrences, while conditional sentences with the keyword “where” is around 150 occurrences.



The most commonly used word in conditional sentences if “if”, in the studies I mostly focus on that kind of sentences.

## Detection

The question immediately arises, why not using some basic search pattern to identify the conditional sentences. It would be enough just to search for a few keywords with the help of regex (we identified above) and we could filter these sentences. However, in practice, it’s not just that simple. There are a plenty of cases, where filter sentences with the keywords results in such sentences, that are not conditionals, but they contain a keyword, as the following example shows:

*“If you are in any doubt about the contents of this Prospectus you should consult your stockbroker, bank manager, solicitor, account or other financial adviser.”*

Even if the sentence itself a conditional sentence, and contains the keyword “if”, we would not think of it as a legal condition. Below there is an example of a “good” example for conditional sentence.

*“If the Issuer is deemed to be a "covered fund", this could significantly impair the marketability and liquidity of the Notes.”*

So, in order to succeed we need to be cautious not to include this kind of sentences into our dataset. Other issue could be that often the conditions are in the previous/following sentence and are separated from the effect.

*“The Dodd-Frank Act requires that federal banking agencies amend their regulations to remove reference to or reliance on credit agency ratings, including but not limited to those found in the federal banking agencies’ risk-based capital regulations. New regulations have been proposed but have not yet been fully implemented in all respects. When such regulations are fully implemented, investments in asset-backed.”*

The first sentence declares the main condition, however in the next sentence which starts with an “if” refers to the first sentence as the condition (such call), and the cause in the second sentence. In this case, we need to tag the two sentences as “conditional” part of the text.

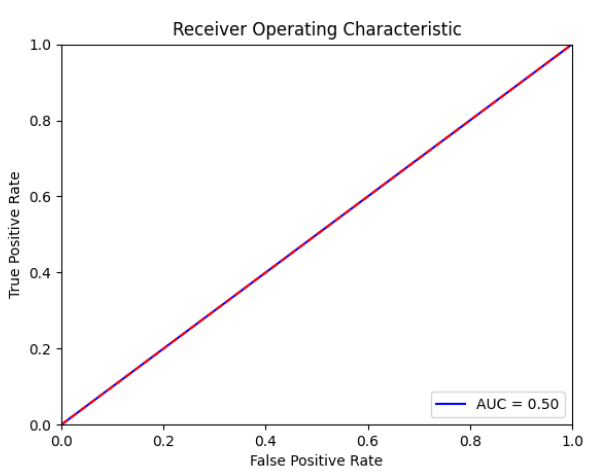
### Classification

Az első ötlet ezeknek a szerkezeteknek a kinyerésére, hogy a paragrafusokat osztályokba sorolom aszerint, hogy tartalmaznak-e feltétlees szerkezet-e vagy sem. Azért paragrafusokat vizsgálunjk, mert mint fentebb is láthattuk, hogy többször több mondatot ölel fel egy-egy logikai rész. Miután azonsoítani tudtuk, hogy igen, ebben a szövegrészben logikai strultúra van, a további feldolgozást már egyéb text mining eszközökkel folytatjuk, melynek célja az less, hogy a folyamatábránkat fel tudjuk építeni

Klasszifikációt a Spacy nevű text mining tool TextCategorizer komponensével fogom végezni. Ehhez először is a spacy által használt “textcat” (TextCategorizer) nevű pipe-ot létre kell hoznunk. Miután létrehoztuk, címkéket is létre kell hoznunk. Jelen esetünkben két címke less, “conditional” és “nonconditional”. A training set összeállítása messze a legnehezebb fealdata. Egy rossz training data jelentősen befolyásolja az eredményt. Mindenekelőtt szüklségünk van olyan paragrafusokra, amelyek tatralmaznak condiiontaleket. De, ahhoz, hogy meg is tudja ezeket különböztetni a többi paragrafustókl, szükségünk van iolyan példákra, mikor nem tartalmaz conditional a paragrafus. A 10 rendelkezésre álló dokumentumban ahogy (Section valami) láthattuk, 1-1 dokumentum rengeteg péládát tartalmaz. Első körben 10-10, 50-50, majd 100-100 pédát gyűjtöttem ki “conditionalre” és nonconditionlre. Minden egyes tanítási kör után egy teszt dokumentumra ráengedve a modellt, szemmel átnéztem az eredményeket, hogy hogyan teljesített a modell. Ezen kívüűl egy roc auc görbét is készítettem, amihez létrehoztam egy 100 péládból álló tesztadatot.

#### 10 Train adat

Miután spacyben ltérehoztam a klasszifikációs modellt, szerettem volna minél előbb validálni magát a koncepciót, így egy 10-10 darabos train-teszt adatom le is teszteltem egyből a modellt. A példákat az első dookumetumből kézzel válogattam ki, egy példa egy paragrafust tartalmaz, amiben van vagy feltételes mód, vagy nincs. Válogatás után egyből lefutattam és a következő roc\_auc score-t kaptam.



Látható, hogy első körös modellunk nem nagyon csináln mást tippelésnél, ezt mutatja a 0.50-es score is. Majd ezután ráeresztettem a “teszt” szerződésemre is az modellt, hogy megvizsgáljam, milyen esetekben téved, illetve milyen esetkre kell különös figeylmet fordítani.

#### 50-500 adat

Most, hogy van már egy rendesen összterakott konfiguráció, már cask jó train adatok kellenek, hogy téynelgesen validájlni tudjuk, vajon a kitalált eljárás működhet-e. Ebben az iterációban már 50-50 példát gyűjö9ttem conditional szerkezetre a dokumentumokban, az eredmény pedig meglehetősen javult. Megnézve a roc-auc görbét, egy 0.74-es eredményt kaptunk.

Ezt tekinthetjük egy nagyon kezdetleges megerősítp tanulási modellnek is (supervised leraning-el keveredve). Megnézve az eredményeket, kigyűjvte azokat a recordokat, ahol hibásan predikált a modellünk, ezekhez hasonló példákat gyűjtöttem. Es

#### 100-100 adat

A mellékelt ábrán látható egy AUC = 0.50 score, ami lélnyegében nem jelent mást, mint, hogy a modellünk tippelt.

As in the Section 2 where we classified the text of the document into paragraphs and titles, we are going to use Spacy’s custom text classifier. The crucial part here is a good training dataset. Ehhez a dokumentumokat át Kellett nézni, majd minden dokumentumból ki Kellett szedni feltételes mondatokat. A 10 doku,entumból 9-ből gyűjtöttem ki ilyen hellegű bekezdéseket. Ahhoz, hogy megfelelően működjön a modell, szükség volt if-es és nem if-es paragrafuisokat is kgyűjteni. Először 5-5 darabbal kezrdtem, és ezzel próbáltam meg felépíteni a modellemet. 1 kategóroát adtam meg spacy-n belül, ez a tag szolgált arra, hogy indikálja, ha egy paragrafuson belül volt-e conditional, majd az utolsó dokumentumból gyűjtött test adatom teszteltem a modellemet.

Első iteráció után nem meglepő, de a modell 50%-os teljesítményt mutatott, tehát a modell cask tippelt

### Named entity recognition

Fentebbi metódusnál láthattuk, hogy sima klasszifikáció nem vezetett komolyabb eredményre, így egy másik megoldáshoz kell folyamodni ahhoz, hogy a feltételes mondatok felismerése sikerhez vezessen, így a spacy által szintén támogatott named entity recognitionhoz fordultam. Named Entity Recognition is a process which deals with identifying and classifying named entities. This named entites usually place, person, organization, time, object, or geographic entity, but in my following approach, I’m going to tag the conditional sentences as named entities.

Spacy has a fast statistical entity recognition system, but we can train it with our own for business specific reasons, the spacy model general performs well for all types of text data.

### Dependency graph

Irodalomjegyzék

1. Levendovszky, J., Jereb, L., Elek, Zs., Vesztergombi, Gy.: Adaptive statistical algorithms in network reliability analysis, Performance Evaluation - Elsevier, Vol. 48, 2002, pp. 225-236
2. National Istruments: LabVIEW grafikus fejlesztői környezet leírása, <http://www.ni.com/> (2010. nov.)
3. Fowler, M.: UML Distilled, 3rd edition, ISBN 0-321-19368-7, Addison-Wesley, 2004
4. Wikipedia: Evaluation strategy, <http://en.wikipedia.org/wiki/Evaluation_strategy> (revision 18:11, 31 July 2012)

Függelék