KİMYA

CHEMISTRY

DOI: http://www.doi.org/10.36719/2707-1146/20/29-34

Tale Namiq oğlu Talıbov
Odlar Yurdu Universitetinin
magistrantı
tale.talibov@adpu.edu.az

TƏBİİ DİL MƏTNLƏRİNDƏN İNFORMASİYANIN ƏLDƏ OLUNMASI ÜÇÜN ONTOLOGİYA İŞLƏMƏLƏRİ

Xülasə

Təbii dil mətnlərinin emalı və ya orijinal adı ilə Natural Language Processing (NLP), süni intellektin inkişafı və dilçiliklə birgə edilmiş araşdırmalar nəticəsində həyatımıza daxil olmuş bir termindir. Başqa cür desək, təbii dil mətnlərinin emalı Azərbaycanca, Rusca, İngiliscə kimi təbii dillərdəki mətnlərin, səs dalğalarının kompüter tərəfindən mənimsənilərək, müxtəlif proqramlarda təhlil edilməsi və kompüter mühitinə köçürülməsidir. Hər kəsin bildiyi kimi, təbii dil insanların ünsiyyət və həyatda qalmaq üçün istifadə etdiyi ən əsas xüsusiyyətlərdən biridir.

Açar sözlər : *təbii dil mətnlərinin emalı, süni intellekt, maşın tərcüməsi, semantik etiketlənmə, ontologiya*

Tale Namiq Talibov

Ontological processing on retrieval information from natural language texts.

Abstract

Natural Language Processing (NLP) is a term that has entered our lives as a result of the development of artificial intelligence and research combined with linguistics. In other words, the processing of natural language texts is the computer adoptering of texts and sound waves into the natural languages such as Azerbaijani, Russian and English, their analysis in various programs and their transfer to the computer environment.

As we all know, natural language is one of the most important features that people use to communicate and survive. Similarly, speaking is a feature of language that occurs in all areas of our lives and is easier to express than writing. In fact, people can control all their work with voice and text. The processing of natural language texts, including our lives, will give us many advantages in doing all our work.

Key words: *Natural Language Processing , Machine Translation, Word Processing, Text Processing, Text Summarization, Argument Aggregation*

Giriş

Kompüterlərin həyatımızın ayrılmaz və vacib hissəsinə çevrilməsi ilə elektron şəkildə saxlanılan məlumatların həcmi əhəmiyyətli dərəcədə artmışdır. Məlumatların elektron mühitdə saxlanmasından sonra bu məlumatların tam və asan şəkildə əldə edilməsini təmin etmək və ya elektron informasiyanı kompüterlər vasitəsilə emal etməklə insanlara müəyyən nəticələr vermək lazımdır. Ona görə də elektron mühitdə saxlanılan məlumatların idarə olunması çox vacib bir tələb kimi qarşıya çıxır.

İnsanların gündəlik həyatda ünsiyyət üçün istifadə etdikləri azərbaycan, türk, ingilis, alman, çin kimi dillərə təbii dillər deyilir. Təbii dil emalı (Natural language processing-nlp) təbii dillərdən istifadə edərək yazılmış mətnlərin kompüterlər tərəfindən başa düşülən hala gətirilməsi prosesidir. Bu, süni intellekt elminin öyrənilən sahələrindən biridir. Təbii dilin işlənməsi iki mövzuda işləyir: təbii dillərin kompüterlər tərəfindən anlaşılması (natural language understanding-nlu) və kompüterlər tərəfindən təbii dillərin yaradılması (natural language generation-nlg).

Təbii dil emal prosesləri dildən dilə fərqlidir. Kompüter əvvəlcə sözün kökündəki şəkilçilərlə çevrilməsinə baxır, buna leksik deyilir. Bundan sonra cümlədəki sözlərin sırasına görə nə demək olduğunu anlamağa çalışır, buna sintaksis deyilir. Sonra o cümlənin mahiyyətcə nəyi izah etməyə çalışdığına baxır, buna semantik deyilir. Nəhayət, cümlələrin bir araya gələrək nəyi ifadə etmək istədiyinə baxır ki, bu da praqmatikdir. Beləliklə kompüter sözün kökünü ayrıca, sözlərin sırasını, cümlənin və nitqin mənasını ayrı-ayrılıqda araşdıraraq nitqin kontekstini öyrənir və bu nitqdən bir məna çıxarır.

1. Təbii dil mətnlərinin emalının elmi nəzəri məsələləri.

Mətnlərin kompüterlər tərəfindən mənalı olması üçün bəzi ön emal tələb olunur. Bu prosesləri aşağıdakı kimi sadalaya bilərik.

- Tokenləşdirmə prosesi
- Mətndəki bütün böyük hərflərlə yazılmış sözləri kiçik hərflərə çevirmək
- Mətndən durğu işarələrinin silinməsi (hashtag, emoji, istifadəçi adı kimi məlumatlar sizin üçün vacib deyilsə, bu məlumatları mətndən silinməsi)
- Mətndən rəqəmlərin müəyyən edilməsi və çıxarılması
- Mətndən dayanacaq sözlərin silinməsi
- Söz köklərinin tapılması
- Mətnin vektora çevrilməsi

Belə bir problem var ki, biz hazırda bütün təbii dillərini və insanların düşüncələrini tam olaraq modelləyə bilmirik. Müasir təbii dil mətinlərinin emalı əsas etibarı ilə xüsusi tapşırıqlar üçün verilən bazasından öyrənmə, ona uyğun riyazi model qurma və modelin yekun qiymətləndirilməsinə əsaslanır. Xüsusi tapşırıqlarım hər biri üçün uyğun verilənlər bazası və riyazi modellər vacibdir. Əksər tapşırıqlarda işarələnmiş verilənlər lazımlıdır, ancaq bu verilənlər bəzi hallarda internetdəki məlumat bazalarında ya mövcud olmur, ya da istənilən şəkildə olmur.

Kompüterin verilənlər toplusunu anlaması üçün mətni riyazi bir anlayış ilə ifadə etməyimiz vacibdir. Təbii dil mətnlərinin emalında bu proses mətnin vektorizasiyası adlanır. Yəni biz yazıdakı hər sözü, cümləni, paraqrafı vektor ilə ifadə edirik. Tək-tək işarələr məna ifadə etmədiyindən bəzi işarə əsaslı neyron modelləri çıxmaq şərti ilə əsasən Təbii dil mətinlərinin emalında ən kiçik vahid kimi ard-arda gələn işarələr toplusunu qəbul edirik. Təbii dil mətinlərinin emalında belə toplu token adlanır. Verilənlər toplusundakı bütün fərqli tokenlərdən ibarət olan set lüğət adlanır.

Mətnin ən sadə vektorizasiyası unitar kod üsuludur. Belə ki, belə vektorların ölçüsü hər bir token üçün lüğətin ölçüsünə bərabərdir və tokenin lüğətdəki yerini ifadə edən indeks 1 ilə, qalan bütün indekslər 0 ilə ifadə olunur.

Təmizlik əsasən internetdən əldə etdiyimiz yazılı məlumatların normal cümlə strukturu halına salmaq üçün istifadə olunur. Bu metod üçün əsasən regex ifadələrindən istifadə olunur.

Təbii dilin emalı ilə bağlı problemlərin müəyyən dərəcədə aradan qaldırılması qabaqcıl obyekt yönümlü proqramlaşdırma dili vasitəsilə mümkündür. Bu nöqtədə Python bizə kifayət qədər geniş kitabxanalar təklif edir. NLP əməliyyatları üçün bəzi ümumi Python kitabxanaları bunlardır:

- ✓ Natural Language Toolkit (NLTK)
- ✓ spaCy
- ✓ TextBlob
- ✓ Gensim
- ✓ pattern
- ✓ polyglot
- ✓ PyNLPI
- ✓ CoreNLP

Kompüterlərin danışdığımız dili başa düşə bilməsi üçün biz çevirmə prosesi edirik. Bu prosesə başlamazdan əvvəl mətni müəyyən bir formata gətirmək lazımdır. Biz bunu mətnin ilkin işlənməsi mərhələlərində həyata keçiririk.

Tokenləşdirmə Prosesi *(Tokenization):* Mətnin əvvəlcədən işlənməsinin ilk addımı, xam mətni kiçik parçalara bölmək təcrübəsi kimi tanınan tokenləşdirmədir. Əsasən iki növ tokenizasiya var. Bunlardan birincisi cümlələr, ikincisi isə sözlərin işarələnməsi üçündür.

Word Tokenizer: Cümldəki sözlərə və durğu işarələrini ayırır.

```
C:\Program Files\Python310\python.exe
>>> import nltk
>>>
>>> metn = "Qarabağda qızıl, gümüş, mis, əlvan metallar, dəmir, sink, qranit, mərmər, qiymətli
daşlar, odadavamlı gil və s. faydalı qazıntılara rast gəlmək olar."
>>>
>>> tokens=nltk.word_tokenize(metn)
>>>
>>> tokens
['Qarabağda', 'qızıl', ',', 'gümüş', ',', 'mis', ',', 'əlvan', 'metallar', ',', 'dəmir', ',', '
sink', ',', 'qranit', ',', 'mərmər', ',', 'qiymətli', 'daşlar', ',', 'odadavamlı', 'gil', 'və',
 's.', 'faydalı', 'qazıntılara', 'rast', 'gəlmək', 'olar', '.']
>>>
```

Mətn Təmizləmə *(Text Cleaning)*

Mətni hissələrə böldükdən sonra araşdırmamız üçün lazım olmayan bəzi durğu işarələri və xüsusi simvolların ortaya çıxdığını görürük və bunların təmizlənməsinə ehtiyac yaranır.

```
C:\Program Files\Python310\python.exe
>>> import string
>>>
>>> message="Qarabağda qızıl, gümüş, mis, əlvan metallar, dəmir, sink, qranit,
 mərmər, qiymətli daşlar, odadavamlı gil və s. faydalı qazıntılara rast gəlmək
 olar."
>>>
>>> print(message.translate(str.maketrans("","",string.punctuation)))
Qarabağda qızıl gümüş mis əlvan metallar dəmir sink qranit mərmər qiymətli daş
lar odadavamlı gil və s faydalı qazıntılara rast gəlmək olar
>>>
```

Bu nümunədə əsas məqsədimiz durğu işarələri və xüsusi simvolların mətndən təmizləməkdir.

Mətnin Normallaşdırılması (*Text Normalization*)

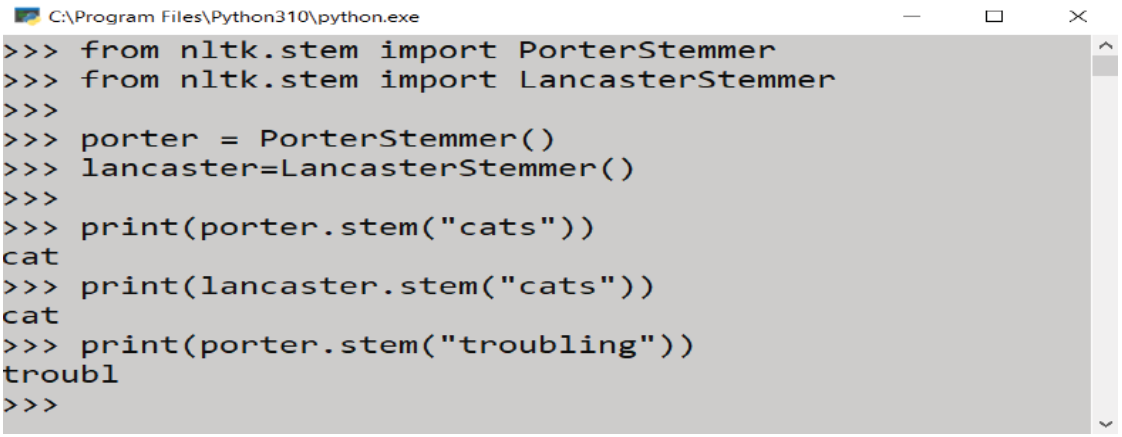
Daha sonra mətni normallaşdırmaq zərurəti önə çıxır. Amma bu nə deməkdir? Normallaşma prosesi müxtəlif kontekstlərə malik ola bilər. Onlardan bəziləri kimi aşağıdakıları göstərmək olar:

- Təkrarlanan boşluqların və və durğu işarələrinin silinməsi.
- Vurğuların çıxarılması. Məsələn, nəzərdən keçirdiyimiz mətndə müxtəlif dillərdə vurğunu göstərən bəzi simvollar varsa, bu ifadələrin silinməsi kodlaşdırma zamanı baş verə biləcək potensial səhvləri aradan qaldırmağa kömək edə bilər.
- Böyük hərflərin çevrilməsi. NLP tədqiqatında yalnız kiçik hərflərlə işləmək fərqli üstünlüklər yarada bilər. Digər tərəfdən, bəzi hallarda böyük hərflər də ad və yer kimi məlumatları çıxarmaq üçün istifadə edə bilərik.
- Xüsusi simvolların və emojilərin çıxarılması və ya çevrilməsi. Məsələn, *Twitter*-dən verilənlərlə işləyərkən tədqiqatın kontekstindən asılı olaraq "hashtag"-ların silinməsi lazım ola bilər.
- Yazı ilə ifadə edilən ədədlərin rəqəmlərlə yazılması. Məsələn, "iyirmi üç" mətninin 23 kimi göstərilməsi tədqiqatda zərurət kimi görünə bilər.
- İxtisarların açıq formalarının göstərilməsi. Məsələn, bir araşdırma ABŞ-ın Amerika Birləşmiş Ştatları kimi abbreviaturasını tələb edə bilər.
- Tarix formatlarını, sosial təminat nömrələrini və ya oxşar formatda digər məlumatları standart formatda yazmaq.
- Orfoqrafiya səhvlərinin düzəldilməsi və sözün müxtəlif variantlarda ifadə olunmasının qarşısının alınması.
- Nadir sözləri daha ümumi sinonimlərlə əvəz etmək.

Bütün bu normallaşdırma prosesləri mövcud işin problemi ilə sıx bağlıdır. Verilənlər toplusunun quruluşundan asılı olaraq hansının həyata keçiriləcəyinə qərar vermək lazımdır.

Sözün kökünü alma (*Stemming*) və Lemmatizasiya (*Lemmatization*) : Stemming sözün kökünü götürməyə verilən addır. Stemming NLP tətbiq olunan dilin təbiətinə görə müxtəlif olur.

Stemmming alqoritmasının üç növü var : Snowball Stemmer, Porter Stemmer və Lancaster Stemmer. Hamısı Python-un NLTK kitabxanasında mövcuddur. Porter Stemmer onların ən qədimidir və sadə dillə desək, tapdığı sözlərin ortaq sonluqlarını çıxararaq ortaq kök tapmağa çalışır. Porter Stemmer-in təkmilləşdirilmiş və daha aqressiv versiyası olan Snowball Stemmer – Porter 2 də adlanır. Snowball stemmer daha sürətli işləyir, ona görə də daha çox istifadə olunur. Lancaster Stemmer, əksinə, ən aqressiv alqoritmdir, burada bəzən həqiqətən heç bir məna daşımayan kökləri tapa bilir, lakin əksinə, daha çox dəyişdirilə bilər.



Lemmatizasiya sözləri morfoloji cəhətdən araşdırır. Azərbaycan dilçiliyində lemmatizasiya təhlil mərhələsi vacib rol oynayır. Lemmatizasiya sözün lemmasını müəyyənləşdirərək tək halda təhlil edilə bilən sözün müxtəlif fleksiyaya uğramış formalarının birlikdə qruplaşması prosesidir.

Kompüter dilçiliyində lemmatizasiya sözün lemmasını alqoritmik müəyyənetmə prosesidir. Lemmatizasiya cümlə daxilində sözün nəzərdə tutduğumuz nitq hissəsi olduğunu və ya mənasını düzgün şəkildə müəyyənetmədən asılıdır, hətta bu geniş kontekstdən, bir-birinə yaxın mənalı cümlələrdən, həmçinin böyük həcmli mətnlərdən ibarət ola bilər.

Əhəmiyyətsiz sözlər (*Stop Words*): Mətnin emal prosesinin ilkin mərhələsinin bir hissəsi kimi mətndən əhəmiyyətsiz sözlərin çıxarılmasını nəzərdən keçirə bilər. Bir dildə çox istifadə edilən və mətindən 1 məna çıxardarkən heç bir əhəmiyyət kəsb etməyən sözləri təmizləmək bizə bəzi üstünlüklər verəcək. Azərbaycan dilində bu sözlərə misal olaraq ilə, amma, sən, ki, ci, da, üçün, az, çox , ən və s göstərmək olar.

Termin tezliyi: Unitar kod modeli çox sadə olsa da, yazıdakı tokenlərin tezliyini ifadə edə bilmir. Məsələn, bir nümunəyə baxaq:

Orxan ingilis dilini bilir. O, almanca da bilir.

Burada təmizlik və ilkin prosesdən sonra “bil” tokeni iki dəfə təkrarlanır. Ancaq bizim vektorumuz unitardır, yəni tokenin neçə dəfə təkrarlanmağına baxmayaraq, vektorda 0 və ya 1 dəyərini alacaq. Bu əksikliyi aradan qaldırmaq üçün termin tezliyindən (TF) istifadə olunur. Belə ki, bu vektor üçün hər bir tokenin yazıda olub-olmamasından başqa, neçə dəfə təkrarlandığı da vacibdir. Deməli, bayaq göstərdiyimiz nümunə üçün TF vektoru belə ifadə edə bilərik:

["orxan", "ingilis", "dil", "bil", "alman", "bil"]
{"orxan":1, "ingilis":1, "dil":1, "bil":2, "alman":1}
[1, 1, 0, 1, 0, 1, 2, 0]

Əsasən termin tezliyi vektoru normalizasiya edildikdən sonra işlədilir. Bunun üçün çoxlu üsul mövcuddur. Ən çox istifdə edilən üsul kimi yazıdakı tokenlərin ümumi sayına nisbəti ilə normallaşdırmadır. Bizim nümunəyə nəzər salsaq, yazı 6 (1 + 1 + 1 + 2 + 1 = 6) tokendən ibarətdir. Deməli vektorumuz TF = [1/6, 1/6, 0, 1/6, 0, 1/6, 1/3, 0] olacaq.

Termin tezliyinin unitar koda görə digər üstün cəhətlərindən biri də yazıdakı hər hansı bir tokeni vektor şəklində yox, real ədədlərlə ifadə edə bilməyimizdir. Yəni, göstərdiyimiz nümunədəki yazını d ilə işarə etsək, tf("orxan", d) = 1/6 olduğunu görə bilərik.

Hər hansısa yazıdakı bir token üçün olan dəyər yalnız və yalnız o yazıdakı tokenlərin işlənməsindən və sayından asılıdır. Məsələn, ümumi verilənlər toplusunda çox nadir olan və ixtiyari bir yazıda 1 dəfə işlənən bir token, həmin yazıda 1 dəfə işlənən başqa sözlə eyni TF dəyərinə malikdir. Ancaq, bu nadir token işləndiyi yazının əsas mənasını ehtiva edə bilər. Bu problemi aradan qaldırmaq üçün tərs dokument tezliyi (IDF) üsulundan istifadə edilir. Bir token üçün IDF dəyəri verilənlər toplusundakı yazıların ümumi sayının həmin token işlənən yazıların sayına nisbətinin loqarifmik dəyərinə bərabərdir.

$$\text{idf}(token, D) = \log \frac{|D|}{|\{d \in D : token \in d\}|}$$

Burada D verilənlər toplusundakı yazıların/dokumentlərin siyahısıdır. Hesablamalar zamanı TF və IDF dəyərlərinin hasilindən istifadə edirik və bu dəyərə **TF-IDF** dəyəri deyilir.

$$\text{tf\_idf}(token, d, D) = \text{tf}(token, d) * \text{idf}(token, D)$$

Yazının TF-IDF vektorunu almaq üçün TF vektorunu vektordakı tokenlərin uyğun IDF dəyəri ilə hasilini tapmaq kifayətdir.

Bəzən söz birləşmələri və ya ardıcıl tokenlər də birlikdə önəmli məna kəsb edə bilirlər. Yazıdakı belə n ardıcıl tokendən ibarət söz birləşmələri n-gram adlandırılır. Tək tokenlərə unigram deyilir. Daha böyük ölçülü birləşmələrdə isə gram ifadəsinin əvvəlinə müvafiq ədədin latınca mənasını əlavə etməklə deyilir (2 : bigram, 3 : trigram).

Nəticə

Ontologiyalar verilənlərin mənasını kompüterlərin idarə edə biləcəyi şəkildə təqdim etməklə, verilənlərin daha ağıllı şəkildə idarə olunmasına imkan verir. Ontologiyaların istifadəsi daha çox işlək məlumatların kompüterə uyğun şəkildə təqdim edilməsini tələb edir. Kompüterin düzgün müəyyən edilmiş verilənlər üzərində çıxaracağı nəticə ilə gizli (yəni açıq şəkildə ifadə olunmayan) informasiyaya çatmaq mümkün olacaq. Təbii dildə yazılmış mətnləri ontologiyalardakı anlayışlarla əlaqələndirərək kompüterlər bu mətnlərdəki anlayışlar arasındakı əlaqələr haqqında biliyə sahib ola bilər və beləliklə, bu mətnlər üzərində daha ağıllı əməliyyatlar həyata keçirərək insanlara fərqli və ya daha keyfiyyətli xidmətlər göstərə bilər.

Ədəbiyyat

1. Maşın tərcüməsində semantik təhlilin ortaq ünsiyyət dilinin yaradılmasına təsiri və əhəmiyyəti // Türkoloji Elmi-Mədəni Hərəkatda Ortaq Dəyərlər və Yeni Çağırışlar" mövzusunda Beynəlxalq Konfrans, – Bakı: Elm və təhsil, – 14-15 noyabr – 2016, – s. 249-252
2. Adalı, E. (2012). Doğal Dil İşleme (Natural Language Processing). Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 2012. 6(6)
3. Seker, S. E. (2015). Metin Madenciliği(Text Mining), YBS Ansiklopedi v. 2, is. 3, pp. 30-32.
4. Evolutionary Process of the Computational Linguistics in Azerbaijan // – USA: The USA Journal of Applied Sciences. CIBUNET (ORT/Publishing), – 2015. № 6, – p. 12-14
5. Chen, X., Liu, Z. ve Sun, M. (2014). A unified model for word sense representation and disambiguation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1025-1035).
6. Matsuo, Y. and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information", International Journal on Artificial Intelligence Tools, Vol. 13, No. 01, pp. 157–169, 2004
7. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", Advances in neural information processing systems, pp. 3111–3119, 2013.

8. McDowell L, Cafarella M. Ontology-driven information extraction with ontosyphon. The Semantic Web-ISWC 2006. 2006;:428–444
9. Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In Proceedings of the Conference on Computational Natural Language Learning (CoNLL-2004).
10. H. D.-G. C. Vlad-Sebastian IONESCU, « "Natural Language Processing and Machine Learning Methods for Software Development Effort Estimation,» Babeş-Bolyai University, 2017.
11. Oelke, D., Momtazi, S., and Keim, D. A. (2012). Natural language processing for text visualization. Tutorial at VisWeek 2012. 118, 125
12. Mayer, T., Rohrdantz, C., Butt, M., Plank, F., and Keim, D. A. (2011). Visualizing vowel harmony. Linguistic Issues in Language Technology, 4(1). 117

Rəyçi : i.ü.f.d, dos. S.Əliyev

Göndərilib: 01.04.2022 Qəbul edilib: 04.05.2022