ҒТАМР 20.23.19 DOI: https://doi.org/10.26577/JMMCS.2020.v107.i3.07

**О.А. Баймуратов, Д.А. Аязбаев***
Сулейман Демирель атындағы университет, Қаскелең қ., Қазақстан
*e-mail: Dauren.Ayazbayev@sdu.edu.kz

# МАМАНДАНДЫРЫЛҒАН СӨЗДЕРДІҢ ВЕКТОРЛАРЫ АРҚЫЛЫ СӨЗДЕРДІҢ ЛЕКСИКАЛЫҚ ТІРКЕСУЛЕРІН АНЫҚТАУ

Сот жүйесінде іс қағаздардың ұйымдастырылуына хатшы жауапты болады. Хаттамаларда қате болған жағдайда, келіспеушілік пайда болуы мүмкін. Сондықтан сөздердің дұрыс лексикалық тіркесуі маңызды. Бұл жұмыста ұйқаспайтын сөздерді табу үшін сөздердің лексикалық тіркесулері есептелінді. Сөздердің лексикалық тіркесулері Skip-gram моделімен анықталды. Skip-gram моделі сөздерді векторлармен сипаттайды. Бұл модельде мағынасы жағынан жақын сөздердің және бір-бірімен лексикалық тіркесетін сөздердің векторлары шамамен бір бағытта болулары керек. Сондықтан екі сөздің бір-бірімен лексикалық тіркесуін анықтау үшін сол сөздердің векторларының арасындағы бұрыштың косинусы есептелінді. Косинустың мәні 1-ге жақындаған сайын екі сөздің лексикалық тіркесулері жоғарлайды. Керісінше, косинустың мәні -1-ге жақындаған сайын екі сөздің лексикалық тіркесулері төмендейді. Бұл жұмыста Қазақстан Республикасының конституциясының бабының мәтініне жаңа сөз енгізген кезде, авторлардың жүйесі енгізілген сөзді табу керек еді. Жүйе кейбір сөздер үшін жоғары дәлдікті көрсеткенімен, кейбір сөздерде қателіктер табылды. өйткені енгізілген жаңа сөз конституцияның бабына қатысты болмағанымен, көрші сөзбен басқа мәтіндерде тіркесе алады. Мысалы, компьютер сөзі мағынасы жағынан конституцияның бабына қатысты болмағанымен, бұл сөз бұрынғы сөзімен лексикалық тіркесе алады. Берілген жұмыс "Отандық білім беруді модернизациялау жағдайында көптілді IT маманының құзыретті инновациялық моделін әзірлеу және енгізу"атты гранттық жоба аясында жүзеге асырылынып жатыр.

**Түйін сөздер**: сөздің векторы, Skip-gram моделі, сөздердің лексикалық тіркесулері.

О.А. Баймуратов, Д.А. Аязбаев*
Университет имени Сулеймана Демиреля, г. Каскелен, Казахстан
*e-mail: Dauren.Ayazbayev@sdu.edu.kz

**Определение лексической сочетаемости слов по векторам специализированных слов**

В системе суда секретарь является ответственным за заполнение протоколов. Маленькая ошибка может привести к недопониманию между людьми. Поэтому секретарь должен стараться не допускать каких-либо ошибок. В данной работе был выполнен анализ слов по их лексической сочетаемости. Лексическая сочетаемость слов была определена по модели Skip-gram. Модель Skip-gram представляет слова в виде векторов. В модели Skip-gram векторы слов, имеющие схожий смысл и лексические сочетаемые слова должны иметь приблизительно одинаковое направление. Поэтому чтобы вычислить лексическое сочетаемость двух слов был определен косинус угла между соответствующими векторами. Если два слова лексический сочетаемы друг с другом, то значение косинуса должен быть приблизительно равным 1. В противном случае, значение косинуса должен быть примерно равным -1. В данной работе в качестве тестирования был взят текст статьи конституции Республики

Казахстан. Когда авторы вводили слова не связанные с контекстом, их система должна была определить введенные слова. Система для некоторых слов показала высокую, а для некоторых слов низкую точность. По мнению авторов, это связано тем, что, несмотря на то, что введенные слова не были связаны с контекстом, они были лексический сочетаемы с соседними словами. Например, слово компьютер по смыслу не был связан с текстом конституции, но это слово может употребляться со словом бұрынғы казахского языка. Данная работа выполняется в рамках грантового проекта Министерства Образования и Науки Республики Казахстан "Разработка и внедрение инновационной компетентностной модели полиязычного IT-специалиста в условиях модернизации отечественного образования".

**Ключевые слова**: векторы слов, модель Skip-gram, лексическое сочетаемость слов.

O.A. Baimuratov, D.A. Ayazbayev
Suleyman Demirel University, Kaskelen, Kazakhstan
*e-mail: Dauren.Ayazbayev@sdu.edu.kz

**Identifying lexical compatibilities of words by vectors of specialized words**

In court system secretary fills protocols. Filling protocols with mistakes can lead to misunderstanding between people. Hence it is important writing protocols properly. In current work to identify mistakes lexical compatibilities of words were computed. To do it Skip-gram model was applied. In Skip-gram model words are represented by vectors. Words with similar meaning and lexically compatible words should have approximately the same direction. Therefore to calculate lexical compatibility of two words cosine value of angle between corresponding two vectors was identified. Cosine value of highly lexically compatible words should be approximately equal to 1. Lexically incompatible words should approximately have value -1. To test their system authors used the text of article of the constitution of the Republic of Kazakhstan. Particularly, words which are not related to meaning of article of the constitution were inserted, and the system had to identify that inserted words. The system for some words showed high accuracy, however some words showed low accuracy. By authors' opinion, it happened because even inserted words were not related in meaning, they could be lexically compatible with their neighbors. For example, word computer can be used in other contexts with word бұрынғы (old) of Kazakh language. This research is carried out within the framework of the Ministry of Education and Science of Republic of Kazakhstan grant project "Developing and implementing the innovative competency-based model of multilingual IT specialist in the course of national education system modernization".

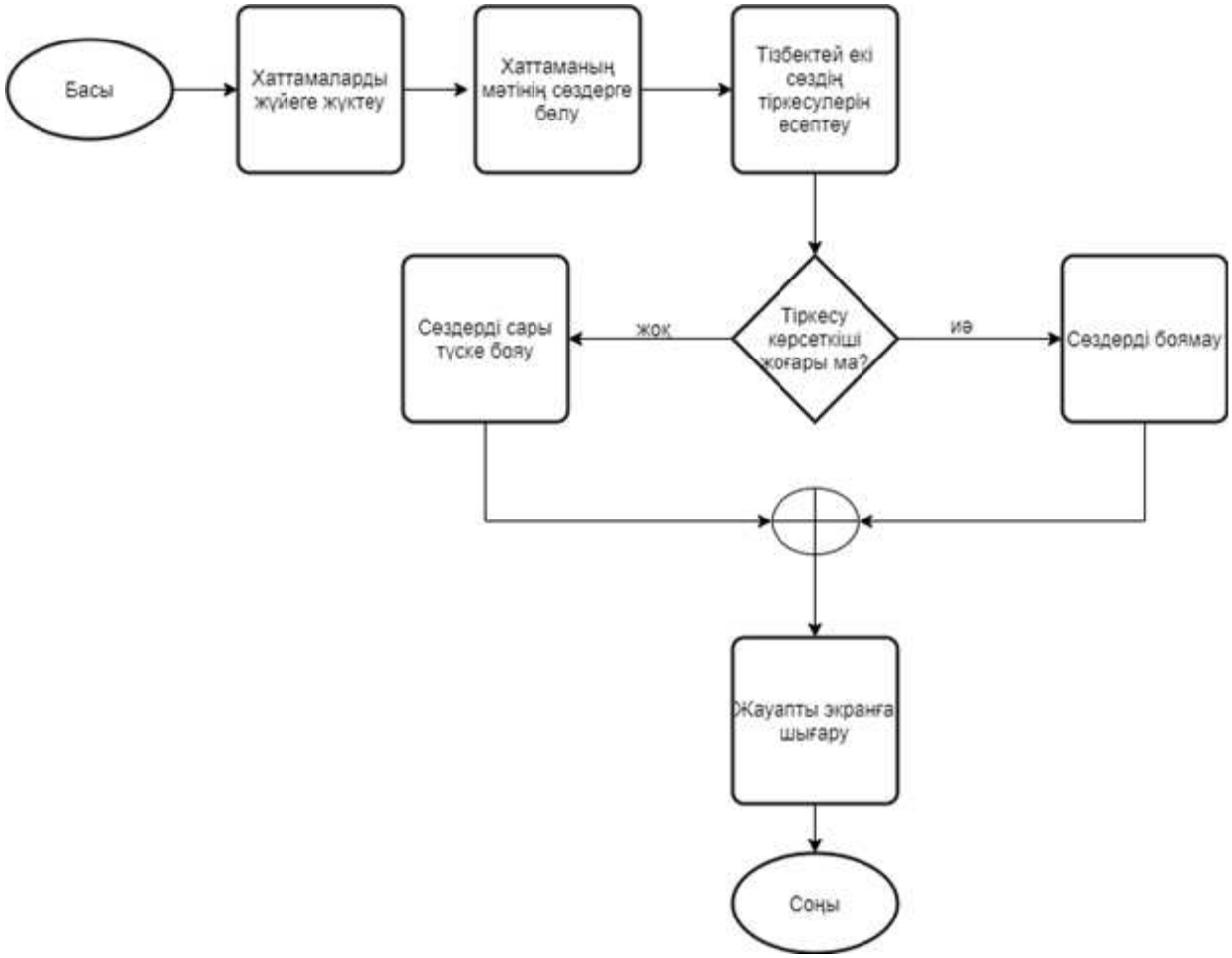
**Key words**: vectors of words, Skip-gram model, lexically compatibilities of words.

## 1 Кіріспе

Қандай жұмыс болмасын, ол мұқияттылықты қажет етеді. Мысалы, дәрігер науқас адамға дәріні тағайындағанда, құрылысшылар үйді салғанда, сот шешімін шығарғанда және т.б. Сот жүйесінде іс қағаздардың ұйымдастырылуларын хатшыға тапсырылынады. Хатшы қандай да бір құжатты қате толтырған жағдайда, оның салдары келіспеушілікті тудырту мүмкін. Бұл жұмыста біз сот жүйесіндегі хатшының хаттамаларды толтыруға көмектесетін қосымшаны даярлағымыз келеді. Біздің қосымшамыз сөздердің лексикалық тіркесуін анықтау керек. Ол үшін біз word embedding әдісін пайдаландық. Word embedding сөздерді векторларға айналдыратын әдіс. Word embedding-те векторлар координаталармен сипатталады. Мағынасы жағынан жақын сөздер шамамен бір бағытта болу керек. Сонымен қатар, word embedding-те сөздің векторының координатасы анықталғанда, сол сөздің басқа сөздермен лексикалық тіркесуі ескеріледі.

## 2 Әдебиетке шолу

Жобамыздың блок-сызбанұсқасы 1-суретте көрсетілді.




1-**сурет** - Жобаның сызба-нұсқасы

1-суретте көрсетілгендей, жүйенің жұмысы хатшының хаттамаларды жүктеуімен басталады. Екі сөздің лексикалық тіркесуін есептеу үшін, сол сөздердің векторларын білу қажет. Сондықтан жүйеде сөздердің векторларынан тұратын сөздік бар.

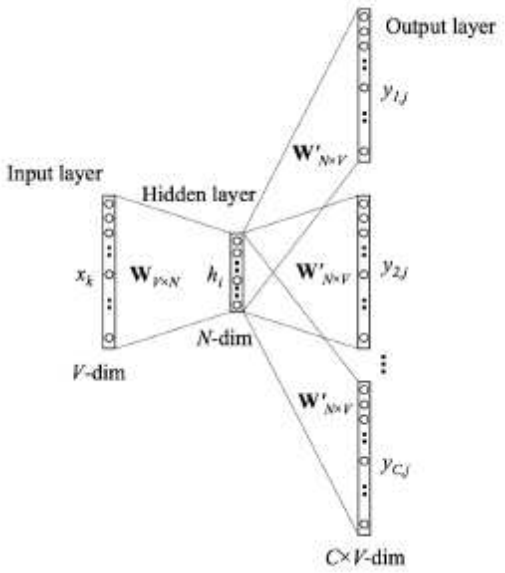
Word embedding-ті әртүрлі салада пайдалануға болады. Мысалы: [1] авторлары word embedding-ті адамдардың пікірлерін білу үшін пайдаланса, [2] авторлары кітаптың ішіндегі кейіпкерлердің өзара байланысын, яғни әлеуметтік желісін (social network) анықтау үшін қолданды.

## 3 Материалдар мен әдістер

Сөзді векторға айналдыратын бірнеше модельдер бар. Мысалы: Skip-gram, continuous bag of words, GloVe. Бұл жобада сөздің векторын анықтау үшін Skip-gram моделі пайдаланылды. Skip-gram моделі берілген сөздің көрші сөздерінің мәнмәтін ішінде кездесу ықтималдығын анықтайды. Skip-gram моделі сөзді векторға айналдыру үшін нейрондық

желіні пайдаланады [3]. Бұл желіде бір сөздің векторын анықтау үшін келесі қадамдардан өту керек [4–6]

1) Корпустың ішінен анықтайын деп жатқан вектордың сөзі кездесетін сөйлемдерді бөліп шығару. Содан кейін сол сөйлемдерден қайталанатын сөздерді алып тастау керек. Одан қалған сөздер нейрондық желінің енгізу қабаты (input layer) болады. Нейрондық желінің құрылымы 2-суретте көрсетілген.




2-**сурет** - Skip-gram-ның нейрондық желісінің құрылымы

Бұл жерде $x$ – енгізу қабатының нейрондары, $W$ – нейрондардың арасындағы салмақтар, $h$ – жасырын қабатының нейрондары, $y$ – шығу қабатының нейрондары, $V$ – әртүрлі сөздердің саны, $C$ – анықтайын деп жатқан вектордың сөзінің көршілерінің саны(терезенің өлшемі). Енгізу қабатында әр сөзге бір нейрон сәйкес.

2) Іздеуге таңдалған сөздің нейронынан басқа нейрондардың бәрі 0 мәнін қабылдайды. Ал іздеуге таңдалған сөзінің нейроны 1-ге тең болады.

3) Нейрондық желіде барлық салмақтар 0 мен 1 арасында тағайындалады. Енгізу мен жасырын, жасырын мен шығу қабаттарындағы салмақтар әртүрлі бола алады.

4) Енгізу қабатындағы барлық нейрондардың мәндерін енгізу мен жасырын қабаттағы салмақтарға көбейту:

$$h = x^T W. \tag{1}$$

5) Шығу қабатының нейрондарының мәндері келесі формуламен анықталады:

$$u = hW^T, \tag{2}$$

W – жасырын қабатымен шығу қабатының арасындағы салмақтар.

6) Шығу қабатының нейрондарының мәндері softmax функциясымен ықтималдылықтарға айналдырылады. Ол үшін келесі формула қолданылады:

$$p\left(w_{c,j} = w_{o,c} | w_i\right) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum\limits_{j'=1}^{V} \exp(u_{j'})}. \tag{3}$$

Бұл жерде:
$w_{c,j}$ – шығу қабатындағы $c$-мәнмәтіндегі $j$-сөз,
$w_{o,c}$ – шығу қабатындағы $c$-сөз,
$w_i$ – енгізу қабатындағы анықтайын деп жатқан вектордың сөзі.

Нейрондық желіде анықтайын деп жатқан вектордың сөзінің көршілер саны мәнмәтін санын анықтайды. Әр мәнмәтін бір көршіге сәйкес. $y_{c,j}$ – $c$-мәнмәтіннің $j$-сөзінің көрші сөз болуының ықтималдығы.

7) Шығу қабатындағы әр нейронға болжау қателігі есептелінеді:

$$e_{c,j} = y_{c,j} - t_{c,j}. \tag{4}$$

Егер $c$-мәнмәтіндегі $j$-сөз с-мәнмәтіннің көрші сөз болса, $t_{c,j}$ 1-ге тең болады. Қалған жағдайларда 0-ге тең болады.

8) Шығу қабатының сөздерінің барлық қателіктері қосылады:

$$\mathrm{EI}_j = \sum_{C=1}^{C} e_{c,j}, \tag{5}$$

$C$-мәнмәтіннің саны.

9) Нейрондық желіде барлық салмақтар келесі формуламен жаңартылады:

$$w_{i,j}{}^{(new)} = w_{i,j}{}^{(old)} - \alpha\ \mathrm{EI_j}\ h_i. \tag{6}$$

Бұл формулада:
$\alpha$ – үйрену жылдамдығының коэффициенті (learning rate),
$w_{i,j}{}^{(new)}$ – жаңа салмақ,
$w_{i,j}{}^{(old)}$ – ескі салмақ,
$h_i$ – жасырын қабатының нейронының мәні.

10) Нейрондық желінің қателігі төмен болғанша 4-9 қадамдарды қайталау.

Бұл жобада екі сөздің лексикалық тіркесуін анықтау үшін, екі сөздің векторларының арасындағы бұрыштың косинусы есептелінді. Екі сөздің мағыналары бір-біріне жақын болған сайын [7, 8] немесе лексикалық тіркесулері үлкейген сайын, косинустың мәні де үлкейеді. Екі векторлардың арасындағы бұрыштың косинусы келесі формуламен есептелінеді:

$$\cos(a) = \frac{a_1b_1 + a_2b_2 + ... + a_nb_n}{\sqrt{a_1^2 + a_2^2 ... + a_n^2}\ \sqrt{b_1^2 + b_2^2 ... + b_n^2}}. \tag{7}$$

Бұл формулада $n$ – вектордың өлшемі. Бұл жұмыста $n$ 100-ге тең болды.

## 4 Нәтижелер мен олардың талқылануы

Сөздердің лексикалық тіркесулерін тексеру үшін, біз Қазақстан Республикасының Конституциясын таңдадық. 90-баптың 1-тармақшасының сөздерінің арасына жаңа сөзді жазғанда, біздің жүйе сол жаңа сөзді табу керек болды. Жаңа сөзді табу үшін сөздердің

векторларының арасындағы косинустары есептелінді. Енгізілген жаңа сөздің оң жағында және сол жағында сөздер болды. Егер енгізілген жаңа сөзбен оның сол жағындағы немесе оң жағындағы сөздердің косинустары қалған басқа сөздердің косинустарынан ең кіші болса, онда жүйе енгізілген жаңа сөзді тапты деп есептелінді. 1-кестеде жүйенің дәлдігін көруге болады.

1-**кесте** – Жүйенің дәлдігі

| **Сөз** | **Дәлдік** |
|---|---|
| Жасыл | 57.14% |
| Сиыр | 100% |
| Қасқыр | 100% |
| Қалам | 71.43% |
| Компьютер | 57.14% |
| Ғарышкер | 28.57% |
| Көлік | 14.29% |
| Ұшақ | 71.43% |
| Темір | 71.43% |
| Алюминий | 85.71% |

Жүйенің дәлдігін есептеу үшін жаңа сөз 90-баптың 1-тармақшасының әртүрлі сөздердің араларына қойылды. 1-кестеде көрсетілгендей біздің жүйе әртүрлі дәлдікті қайтарды. Тек сиыр, қасқыр сөздері 100% дәлдікті көрсетті. Өйткені бұл сөздер 90-баптың 1-тармақшасының сөздерімен лексикалық тіркес емес(мысалы: Республикалық сиыр, ресми сиыр). Дегенмен кейбір сөздер үшін жүйенің дәлдігі төмен болды. Өйткені сол сөздер мағынасы жағынан 90-баптың 1-тармақшасына сәйкес келмесе де, оң жағындағы немесе сол жағындағы сөзбен лексикалық тіркесе алады. Мысалы: ұшақ деген сөз бұрынғы деген сөзімен, ғарышкер сөзі ресми сөзімен лексикалық тіркесе алады.

## 5 Қорытынды

Жоғарыда көрсетілгендей word embedding-тың өзінің кемшіліктері бар. Мысалы, сөздердің лексикалық тіркесулерін анықтау үшін, конституцияның барлық сөздері сөздікте болу керек.Сонымен қатар, екі вектордың арасындағы бұрыштың косинусы арқылы тек осы екі векторлардың өзара лексикалық тіркесулерін бағалай аламыз. Дегенмен, осы векторлардың сөздерінің лексикалық тіркесулеріне оларға дейін және кейін тұрған сөздер де әсер ете алады. Сондықтан жүйеміздің дәлдігін көбейту үшін, біздің phrase embedding-ты пайдаланғанымыз жөн. Қазіргі таңда әртүрлі тілдер үшін векторлардың бірнеше нұсқалары бар. Олар [9, 10] қол жетімді.

## Әдебиеттер тізімі


[1] И.В. Бондарева, Д.Г. Лагерев. 2018, Исследование методов векторного представления текстовой информации для решения задачи анализа тональности, Всероссийская научная конференция "Информационные технологии интеллектуальной поддержки принятия решений Уфа-Ставрополь, Россия, 2018, 10-15 стр.

[2] Gerhard Wohlgenannt, Ekaterina Chernyak, Dmitry Ilvovsky, 2016, Extracting Social Networks from Literary Text with Word Embedding, Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), December 11-17 2016. pages 18–25.

[3] http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/. Қарау датасы: 10.06.2020.

[4] David Meyer, 2016, How exactly does word2vec work? July 31, 2016. Pages 1-18.

[5] https://hmkcode.com/ai/backpropagation-step-by-step/. Қарау датасы: 10.06.2020.

[6] https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-skip-gram.html. Қарау датасы: 10.06.2020.

[7] Nawal Ould-Amer, Philippe Mulhem, Mathias Géry, Karam Abdulahhad, 2016, Word Embedding for Social Book Suggestion, Clef 2016 Conference, 09.05.2016, Volume 1609

[8] Ensaf Hussein Mohamed, Eyad Mohamed Shokry, 2020, QSST: A Quranic Semantic Search Tool based on word embedding, Journal of King Saud University –Computer and Information Sciences, 4 January 2020

[9] https://code.google.com/archive/p/word2vec/. Қарау датасы: 10.06.2020.

[10] https://sites.google.com/site/rmyeid/projects/polyglot. Қарау датасы: 10.06.2020.


## References


[1] I.V. Bondareva, D.G. Lagerev. 2018, Issledovanie metodov vektornogo predstavlenija tekstovoj informacii dlja reshenija zadachi analiza tonal'nosti, Vserossijskaja nauchnaja konferencija "Informacionnye tehnologii intellektual'noj podderzhki prinjatija reshenij Ufa-Stavropol, Russia, 2018, 10-15 p.

[2] Gerhard Wohlgenannt, Ekaterina Chernyak, Dmitry Ilvovsky, 2016, Extracting Social Networks from Literary Text with Word Embedding, Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), December 11-17 2016. pages 18–25.

[3] http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/. Accessed date: 10.06.2020.

[4] David Meyer, 2016, How exactly does word2vec work? July 31, 2016. Pages 1-18.

[5] https://hmkcode.com/ai/backpropagation-step-by-step/. Accessed date: 10.06.2020.

[6] https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-skip-gram.html. Accessed date: 10.06.2020.

[7] Nawal Ould-Amer, Philippe Mulhem, Mathias Géry, Karam Abdulahhad, 2016, Word Embedding for Social Book Suggestion, Clef 2016 Conference, 09.05.2016, Volume 1609

[8] Ensaf Hussein Mohamed, Eyad Mohamed Shokry, 2020, QSST: A Quranic Semantic Search Tool based on word embedding, Journal of King Saud University –Computer and Information Sciences, 4 January 2020

[9] https://code.google.com/archive/p/word2vec/. Accessed date: 10.06.2020.

[10] https://sites.google.com/site/rmyeid/projects/polyglot. Accessed date: 10.06.2020.