
# Дәріс 13: Табиғи тілді өңдеу (NLP)

Оқытушы: Сарсембаева Т.С.

# Дәрістің мақсаты

NLP-дің не екенін, оның жасанды интеллект саласындағы орнын және күнделікті өмірдегі мысалдарын (Google Search, Translate) түсіндіруден басталады. Содан кейін, NLP-дің негізгі қиындығы – адам тілінің күрделілігін және оны сандық форматқа айналдыру қажеттілігін көрсету қарастырылады. Дәріс барысында студенттер мәтінді алдын ала өңдеудің (text preprocessing) негізгі кезеңдерімен танысады. Сондай-ақ, қарапайым талдау әдісі ретінде "Сөз бұлттарын" (Word Clouds) және оның сөз жиілігіне негізделетінін зерттеу, статистикалық тілдік модельдеуді (LM) және n-грамм (биграмм, триграмм) модельдерін сөздердің реттілігін есепке алу үшін қалай қолданылатынын түсіндіру көзделген. Соңында, тақырыптық модельдеудің (Topic Modeling) не екенін және оның негізгі алгоритмі Latent Dirichlet Allocation (LDA) қалай жұмыс істейтінін меңгерту мақсат етіледі.

## Негізгі сұрақтар:

- Табиғи тілді өңдеу (NLP) дегеніміз не және оның негізгі қиындығы неде?
- Мәтінді алдын ала өңдеу (text preprocessing) не үшін қажет?
- Токенизация, Стемминг және Лемматизацияның айырмашылығы неде?
- "Сөз бұлтын" (Word Cloud) жасау кезінде "тоқтау сөздерді" (stop words) неліктен алып тастау керек?
- n-грамм моделі (мысалы, биграмм) "сөздер жиынтығы" (bag of words) моделінен несімен ерекшеленеді?
- Latent Dirichlet Allocation (LDA) алгоритмінің негізгі идеясы қандай?
- LDA-да Гиббс іріктеуі (Gibbs Sampling) қандай рөл атқарады?

# Табиғи тілді өңдеу дегеніміз не?

Табиғи тілді өңдеу (Natural Language Processing, NLP) – бұл жасанды интеллект (AI), информатика және лингвистика салаларының қиылысында орналасқан, компьютерлерге адам тілін (қазақ, ағылшын, орыс тілдері сияқты) "түсінуге", интерпретациялауға және генерациялауға мүмкіндік беруге бағытталған бағыт.

Біз NLP-ді күнделікті өмірде үнемі қолданамыз:

- Google Search: Сіздің сұранысыңызды түсініп, ең мағыналы жауапты табу.
- Google Translate / Yandex Translate: Бір тілден екінші тілге аудару.
- Дауыстық көмекшілер (Siri, Alexa): Сіздің дауысыңызды танып, айтқаныңызды орындау.
- Спам-фильтрлер: Электрондық поштаның мазмұнына қарап, оның спам екенін анықтау.
- Авто-түзету (Autocorrect): Телефонда мәтін тергенде қателеріңізді түзету.

NLP-дің негізгі қиындығы неде? Адам тілі – өте күрделі құрылым. Ол көп мағыналылыққа (ambiguity), сарказмға, контекстке, мәдени ерекшеліктерге толы. Компьютерлер сандармен жұмыс істеуге дағдыланған, ал мәтінмен емес.

NLP-дің бірінші және ең маңызды қадамы – осы ретсіз, "лас" мәтінді компьютер түсіне алатын құрылымдалған, сандық форматқа айналдыру.

# Мәтінді дайындау: Грамматика, Токенизация және Нормализация

Біз мәтінмен жұмыс істеуді бастамас бұрын, оны "тазалау" және құрылымдау процесінен өткізуіміз керек. Бұл процесс мәтінді алдын ала өңдеу (text preprocessing) деп аталады.

## 1. Токенизация **(Tokenization):**

Бұл – мәтінді кішігірім бөліктерге, яғни токендерге бөлу процесі. Токендер сөйлем, сөз немесе тіпті символ болуы мүмкін.

- Сөйлемге токенизациялау: Мәтінді жеке сөйлемдерге бөлу (әдетте ., !, ? белгілері арқылы).
- Сөзге токенизациялау: Әр сөйлемді жеке сөздерге бөлу.

Мысал:

- Бастапқы мәтін: "Сәлем, әлем! Бүгін NLP оқимыз."
- Сөйлем токендері: ["Сәлем, әлем!", "Бүгін NLP оқимыз."]
- Сөз токендері: ["Сәлем", ",", "әлем", "!", "Бүгін", "NLP", "оқимыз", "."]

## 2. Нормализация **(Normalization):**

Токенизациядан кейін біз мағынасы бір, бірақ жазылуы әртүрлі токендерді біріздендіруіміз керек.

- Төменгі регистрге ауыстыру (Lowercasing): "Кітап" және "кітап" сөздерін бірдей ету үшін барлығын төменгі регистрге (lowercase) ауыстыру. ["сәлем", ",", "әлем", "!", "бүгін", "nlp", "оқимыз", "."]
- Пунктуацияны алып тастау: Егер тыныс белгілері маңызды болмаса (мысалы, спам-фильтрде), оларды алып тастаймыз. ["сәлем", "әлем", "бүгін", "nlp", "оқимыз"]
- Тоқтау сөздерді (Stop Words) алып тастау: "және", "немесе", "бірақ", "мен", "сен", "ол" сияқты жиі кездесетін, бірақ мағыналық жүктемесі аз сөздерді ("тоқтау сөздерді") алып тастау. Бұл мәтіннің негізгі мазмұнын бөліп алуға көмектеседі.

### Сөздің негізін табу **(Stemming & Lemmatization):**

- Стемминг (Stemming): Сөздің түбірін табу үшін жалғауларды "кесіп" тастайтын қарапайым, ережеге негізделген процесс (мысалы, "оқимыз", "оқыды", "оқушы" > "оқ"). Нәтиже әрқашан мағыналы сөз бола бермеуі мүмкін.
- Лемматизация (Lemmatization): Сөзді оның бастапқы, сөздіктегі формасына (леммасына) келтіру үшін морфологиялық талдауды қолданады (мысалы, "оқимыз" > "оқу", "бардым" > "бару"). Бұл стеммингке қарағанда күрделі, бірақ дәлірек әдіс.

Осы процестерден кейін ғана біздің мәтініміз талдауға дайын болады.

# Қарапайым талдау: Сөз бұлттары **(Word Clouds)**

Мәтінді алдын ала өңдеуден кейін жасай алатын ең қарапайым талдау – бұл сөздердің жиілігін (word frequency) есептеу.

Сөз бұлты (Word Cloud) – бұл мәтіндік деректердегі сөздердің жиілігін көрнекі түрде ұсынатын визуализация әдісі.

- Мәтінде жиі кездесетін сөздер үлкенірек және қалың шрифтпен көрсетіледі.
- Сирек кездесетін сөздер кішірек шрифтпен көрсетіледі.

Сөз бұлтын жасау үшін:

1. Мәтінді токенизациялап, нормализациядан өткізу (төменгі регистр, пунктуацияны жою).
2. Тоқтау сөздерді (stop words) міндетті түрде алып тастау. (Егер алып тастамасақ, "және", "мен", "де" сияқты сөздер ең үлкен болып шығады да, визуализацияның мағынасы болмайды).
3. Әрбір қалған сөздің жиілігін санау.
4. Осы жиіліктер негізінде "бұлтты" генерациялау.

Бұл әдіс мәтіннің негізгі тақырыбын немесе "не туралы" екенін бір қарағаннан-ақ жылдам түсінуге мүмкіндік береді.

# Статистикалық тілдік модельдеу: n-граммдық тіл үлгілері

Мәтінді жай ғана "сөздер жиынтығы" (bag of words) ретінде қарастыру (сөз бұлтындағыдай) көп ақпаратты жоғалтады. Сөздердің реттілігі өте маңызды.

"Ит адамды қапты" vs "Адам итті қапты" – сөздер бірдей, бірақ мағынасы мүлдем басқа.

Тілдік модель (Language Model, LM) – бұл сөздер тізбегінің пайда болу ықтималдығын есептейтін статистикалық модель. Оның ең жиі қолданылатын міндеті – берілген алдыңғы сөздерге сүйеніп, келесі сөзді болжау.

Мысалы, "Менің атым..." дегеннен кейін "Айгүл" сөзінің пайда болу ықтималдығы "компьютер" сөзінен әлдеқайда жоғары.

Толық сөйлемнің ықтималдығын $P(\omega_1, \omega_2, .., \omega_n)$ математикалық түрде есептеу өте қиын, себебі тілдегі комбинациялар саны шексіз.

n-грамм (n-gram) моделі бұл мәселені Марков болжамы (Markov Assumption) арқылы жеңілдетеді: "Келесі сөздің ықтималдығы оның алдындағы барлық сөздерге емес, тек n-1 сөзге ғана тәуелді."

- Unigram (1-грамм): $P(\omega_i)$ . Әр сөздің ықтималдығы басқа сөздерге тәуелсіз (бұл "сөздер жиынтығы" моделі).
- Bigram (2-грамм): $P(\omega_i \vee \omega_{i-1})$ . Келесі сөздің ықтималдығы тек бір алдыңғы сөзге тәуелді.

$P$(интеллект $\vee$ жасанды)

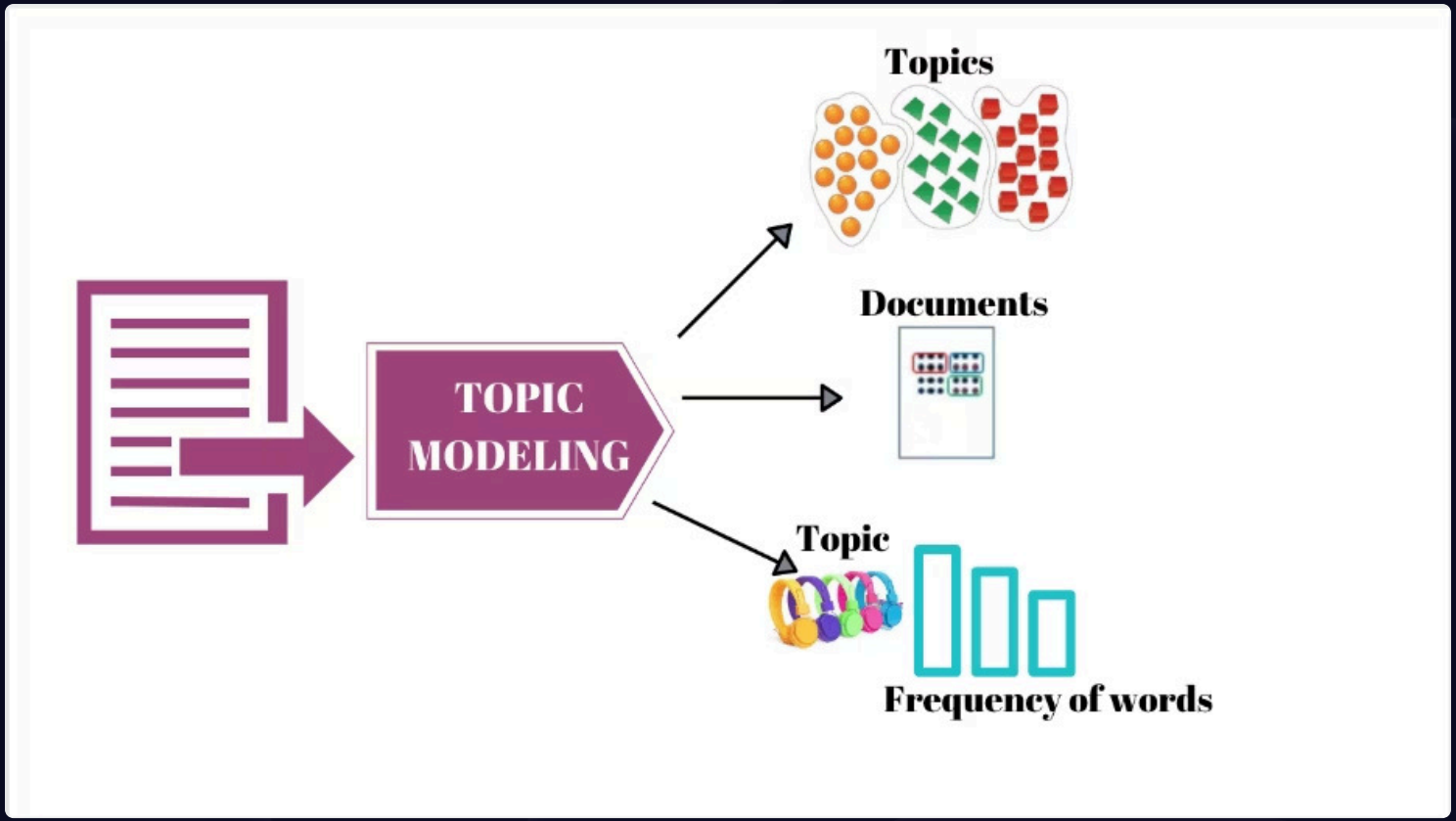
- Trigram (3-грамм): $P(\omega_i \vee \omega_{i-2}, \omega_{i-1})$ . Келесі сөздің ықтималдығы алдыңғы екі сөзге тәуелді.

$P$(оқыту $\vee$ машиналық) $\neq P$(оқыту $\vee$ терең) $\neq P$(оқыту $\vee$ қашықтықтан)

n-граммдар үлкен мәтін корпусындағы (мысалы, барлық Уикипедия) тіркестердің жиілігін санау арқылы үйретіледі.

## Тақырыптық модельдеу **(Topic Modeling)**

Бізде бірнеше құжат (мақалалар, хаттар, жаңалықтар) бар делік. Біз бұл құжаттардың "не туралы" екенін автоматты түрде, алдын-ала белгілемей-ақ (яғни, қадағаланбайтын оқыту) білгіміз келеді.




Тақырыптық модельдеу (Topic Modeling) – бұл құжаттар жиынтығындағы жасырын "тақырыптарды" анықтауға арналған статистикалық әдіс.

Ең танымал алгоритм – Latent Dirichlet Allocation (LDA).

## LDA-ның негізгі идеясы:

LDA екі негізгі болжамға сүйенеді:

1. Әрбір құжат – бұл тақырыптардың араласпасы (mixture of topics). Мысалы, бір жаңалық мақаласы: 70% "Саясат", 20% "Экономика", 10% "Спорт".
2. Әрбір тақырып – бұл сөздердің ықтималдық үлестірімі (a distribution over words). Мысалы, "Спорт" тақырыбы: 30% "футбол", 20% "чемпион", 15% "команда", 10% "ойын"... Ал "Экономика" тақырыбы: 25% "акция", 20% "инфляция", 15% "банк", 10% "несие"...

## LDA қалай жұмыс істейді?

Кіріс: "Сөздер жиынтығы" (BoW) ретінде ұсынылған құжаттар жинағы (мысалы, 1000 мақала) және K (тақырыптар саны, мысалы, 10) саны.

Алгоритм (мысалы, Гиббс іріктеуі): Модель әрбір құжаттағы әрбір сөзді қарап шығып, оны K тақырыптың біріне тағайындауға тырысады. Ол мұны итеративті түрде жасайды, әрбір сөздің тақырыбын оның "көршілес" сөздері мен құжаттарына қарап үнемі жаңартып отырады.

Шығыс: Құжат-Тақырып матрицасы: Әр құжаттың тақырыптық құрамы (мысалы, Doc1: [0.7, 0.2, 0.1, ...]).

Тақырып-Сөз матрицасы: Әр тақырыпты құрайтын негізгі сөздер (мысалы, Topic1: ["футбол", "чемпион", ...]).

Бұл әдіс мыңдаған құжатты тез арада талдап, олардың негізгі мазмұнын түсінуге өте тиімді.

## Ремарка: Гиббс бойынша іріктеу әдісі **(Gibbs Sampling)**

Жоғарыда "LDA алгоритмі" деп айттық. Бірақ LDA шын мәнінде модельдің өзі, ал оны үйрету (яғни, сол жасырын тақырыптарды табу) үшін арнайы алгоритмдер қажет. Гиббс іріктеуі – осы мақсатта қолданылатын ең танымал әдістердің бірі.

Бұл өте күрделі статистикалық әдіс, бірақ оның LDA-дағы жұмыс істеу логикасын қарапайым тілмен түсіндіруге болады:

### Мақсаты:

Әрбір құжаттағы әрбір сөздің қай тақырыпқа жататынын анықтау.

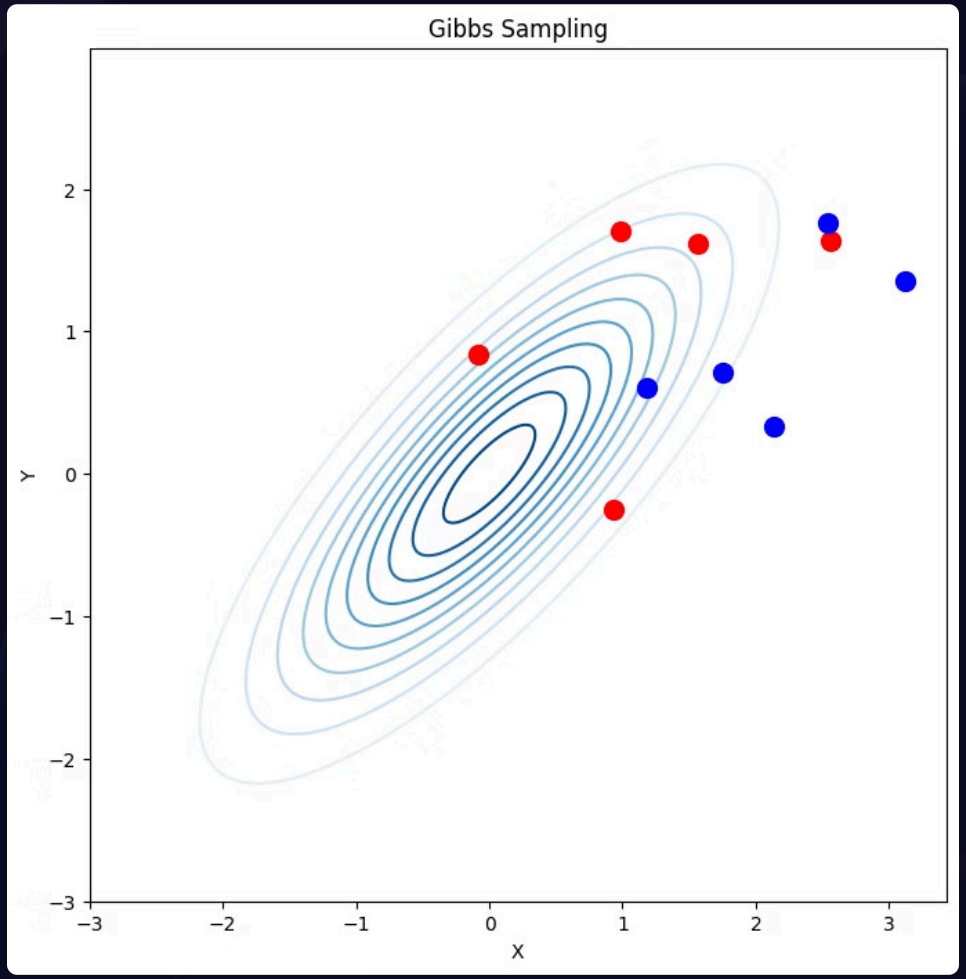
### Процессі:

1. Бастапқы тағайындау: Алдымен, барлық құжаттардағы әрбір сөзге $K$ тақырыптың бірін кездейсоқ тағайындаймыз. (Нәтижесі, әрине, мағынасыз болады).
2. Итеративті жаңарту: Модель мыңдаған рет (итерация) келесі әрекетті қайталайды:

> Ол бір құжаттағы бір ғана сөзді (w) алады.
>
> Оның ағымдағы тақырып тағайындауын "ұмытады" (өшіреді).
>
> Енді ол осы w сөзіне жаңа тақырыпты (t) тағайындау керек. Ол шешімді екі сұраққа негіздеп қабылдайды:
>
> "Бұл құжатқа қандай тақырыптар ұнайды?" Ол осы құжаттағы басқа сөздердің қандай тақырыптарға тағайындалғанына қарайды.
>
> "Бұл сөзге қандай тақырыптар ұнайды?" Ол басқа барлық құжаттарда осы w сөзінің қандай тақырыптарға тағайындалғанына қарайды.
>
> Осы екі ықтималдықты біріктіріп, w сөзіне жаңа тақырыпты (кездейсоқ, бірақ ықтималдыққа сүйеніп) тағайындайды.

3. Тұрақтану: Осы процесс мыңдаған рет қайталанғаннан кейін, сөздердің тақырыпқа тағайындалуы кездейсоқ болмай, нақты құрылымды көрсете бастайды.

Қысқаша айтқанда, Гиббс іріктеуі – бұл LDA-ның "қозғалтқышы", ол сөздерді итеративті түрде қайта-қайта сұрыптау арқылы жасырын тақырыптық құрылымды табады.



## Дәрісті меңгеруге арналған бақылау сұрақтары:

1. Табиғи тілді өңдеу (NLP) дегеніміз не және оның негізгі қиындығы неде?
2. Мәтінді алдын ала өңдеу (text preprocessing) не үшін қажет?
3. Токенизация, Стемминг және Лемматизацияның айырмашылығы неде?
4. "Сөз бұлтын" (Word Cloud) жасау кезінде "тоқтау сөздерді" (stop words) неліктен алып тастау керек?
5. n-грамм моделі (мысалы, биграмм) "сөздер жиынтығы" (bag of words) моделінен несімен ерекшеленеді?
6. Latent Dirichlet Allocation (LDA) алгоритмінің негізгі идеясы қандай?
7. LDA-да Гиббс іріктеуі (Gibbs Sampling) қандай рөл атқарады?

## Әдебиеттер тізімі:


1. Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.
2. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022. http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf
3. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python (NLTK Book). O'Reilly Media. https://www.nltk.org/book/
4. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media. https://github.com/ageron/handson-ml3