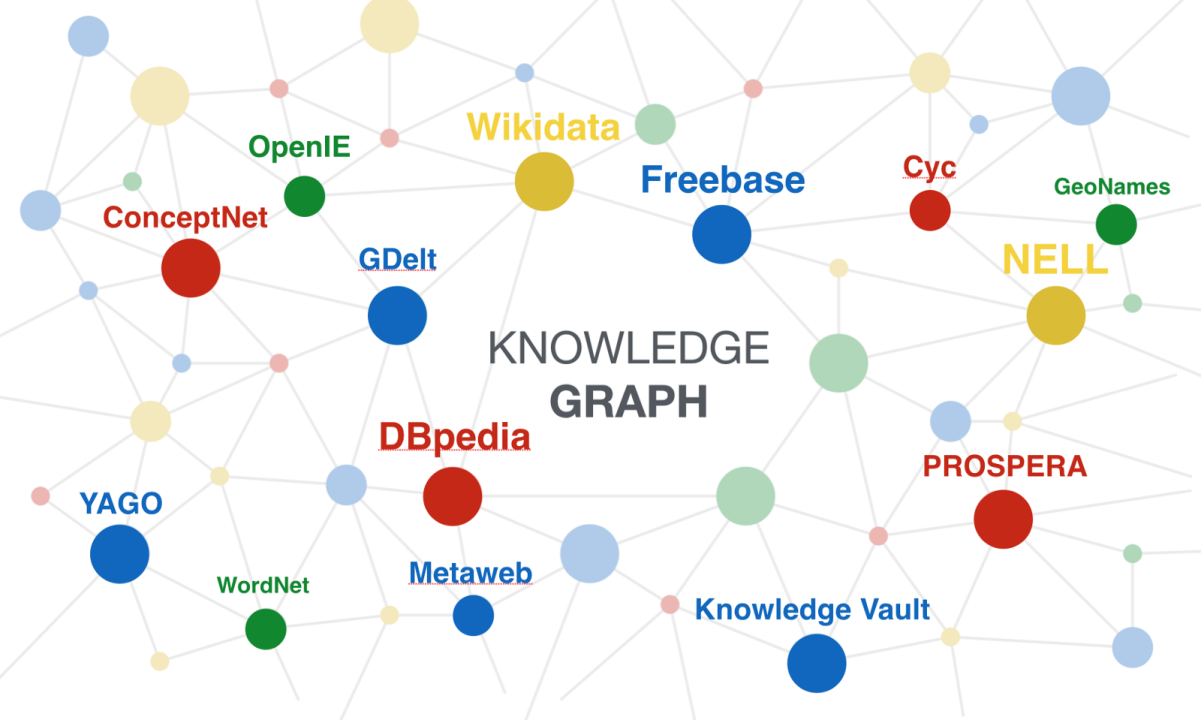
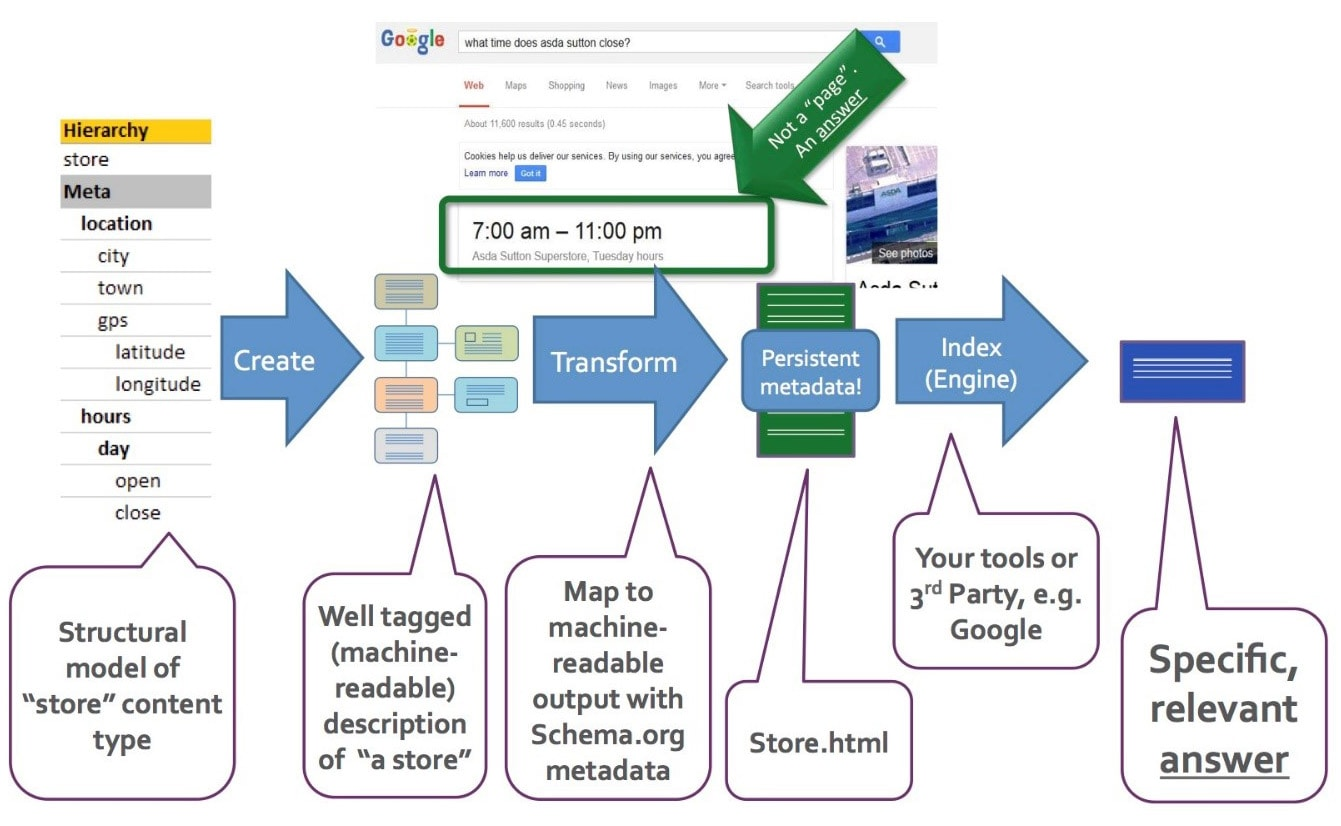
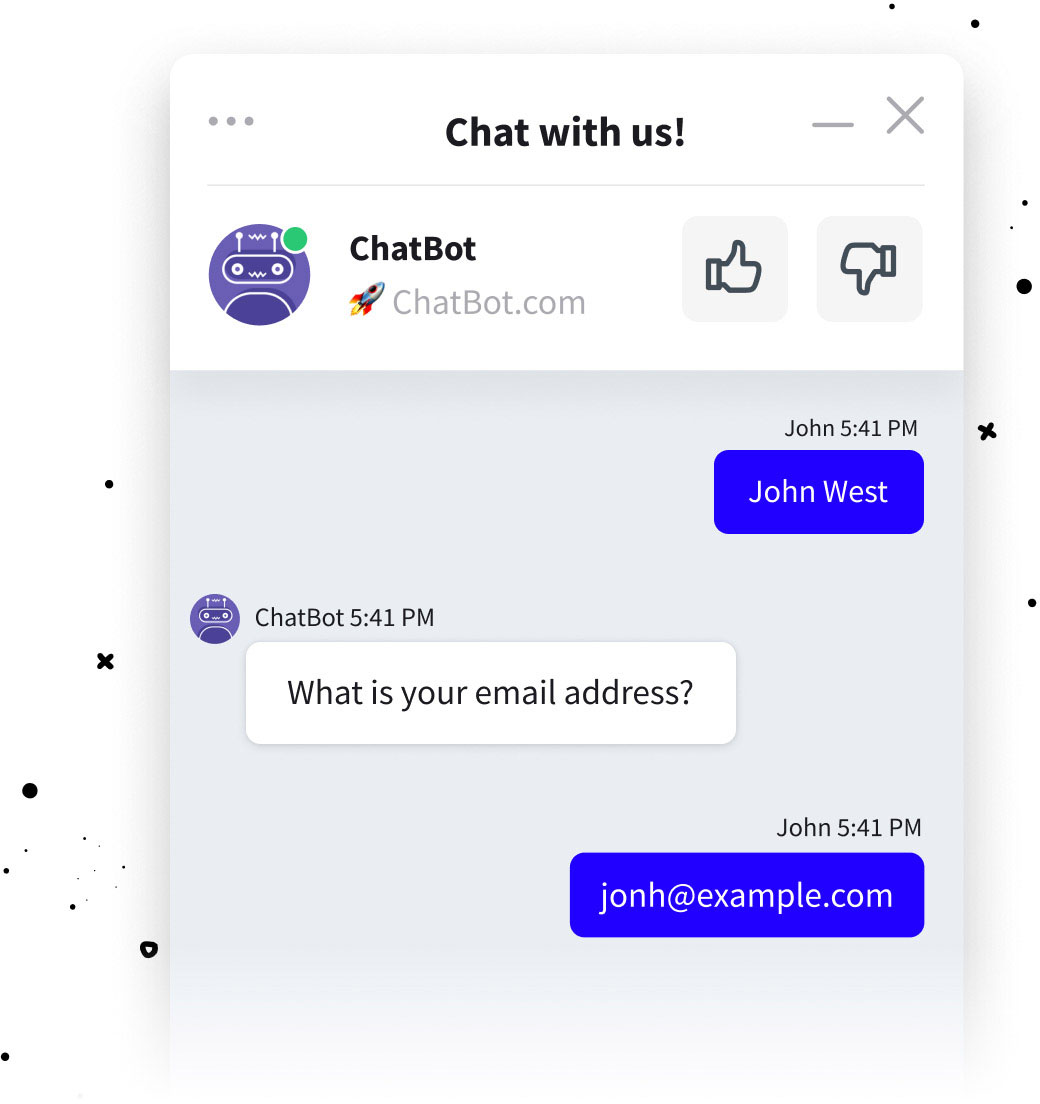
| **PoC Lab – 아이펠톤 프로젝트 계획서** |
| --- |

| **개발아이템명** | using Django and NLP models to make smart search engine and knowledge graph | | |
| --- | --- | --- | --- |
| **소속** | daegu campus | | |
| **신청자 성명** | sun huai | **담당퍼실** | Li |

□ 프로젝트 아이템 개요(요약)

| **아이템**  **소개** | ·using django to make a knowledge db website for intelligent search and graph show and chatbot |
| --- | --- |
| **아이템의 특징 및 차별성** | · christianity field, neo4j graph show, elasticsearch engine, |
| **이미지** |  |





1. 문제인식 (Problem)

**1-1 프로젝트의 목표 및 목적(필요성)**

◦ to make a knowledge db

* gather christianity articles
* clean the data, summarize and get keyword etc

◦ to make intelligent search like google with recommendation

* using elasticsearch for searching with intelligent keyword recommendation

◦ show it in graph visualization

* realize the logical relations of keywords

◦ make chatbot

-understand questions

- reply with knowledge logic

**1-2 아이템의 독창성**

◦ christianity db require christianity knowledge

-can help christians to understand deeply of bible

- can help nonchristians to learn christianity

- use graph method to show christianity knowledge

2. 개발 및 연구 내용

**2-1. 구현 내용 상세(구현 가능성)**

db: pregresql , elasticsearch, neo4j graph DB

haystack: use haystack to make nlp applications and integrated to django

django-haystack: to integrate Elasticsearch with django for search engine

**2-2. 개발 아이템 기대효과**

1. make a website for this project for using to study and search
2. make some apps for mobile end to use

3. 실행 계획

**3-1. 기간내 프로젝트 구현 완성을 위한 전략**

| ◦ use wikipedia data and haystack to test the nlp applications  ◦ represent NLP applications including knowledge base and search engine on Django  ◦ scraping christianity data and show it in website and tune the models |
| --- |

**3-2. 아이펠톤 기간 내 마일스톤**

| **Task** | **목표기간** | **세부내용** |
| --- | --- | --- |
| using wikipedia test data to make nlp applications | 2 weeks | *integrate haystack and models* |
| *django presentation* | *2 weeks* | *django-haystack and neo4j etc show* |
| *preparing christianity data* | *1 week* | *scraping websites* |
| *better the applications and website* | *1 week* | *make changes and repair bugs* |
|  |  |  |

**3-3. 팀장 및 팀원의 역할 분배**

|  |
| --- |

| **순번** | **주요 담당업무** | **역할 상세** | **인원** |
| --- | --- | --- | --- |
| 1 | data preparation |  | sun huai |
| 2 | NLP applications |  | sun huai |
| 3 | django representation |  | sun huai |

4. Reference

**Elasticsearch** is a [search engine](https://www.wikiwand.com/en/Search_engine_(computing)) based on the [Lucene](https://www.wikiwand.com/en/Lucene) library. It provides a distributed, [multitenant](https://www.wikiwand.com/en/Multitenancy)-capable [full-text search](https://www.wikiwand.com/en/Full-text_search) engine with an [HTTP](https://www.wikiwand.com/en/HTTP) web interface and schema-free [JSON](https://www.wikiwand.com/en/JSON) documents. **url:**[**https://www.elastic.co/**](https://www.elastic.co/)

**Neo4j** is a [graph database](https://www.wikiwand.com/en/Graph_database) [management system](https://www.wikiwand.com/en/Management_system) developed by Neo4j, Inc. Described by its developers as an [ACID](https://www.wikiwand.com/en/ACID)-compliant transactional database with native graph storage and processing,[[3]](https://www.wikiwand.com/en/Neo4j#citenoteproductdetails3) url:<https://neo4j.com/>

**Haystack** is an **open-source framework** for building **search systems** that work intelligently over large document collections. Recent advances in NLP have enabled the application of question answering, retrieval and summarization to real world settings and Haystack is designed to be the bridge between research and industry.

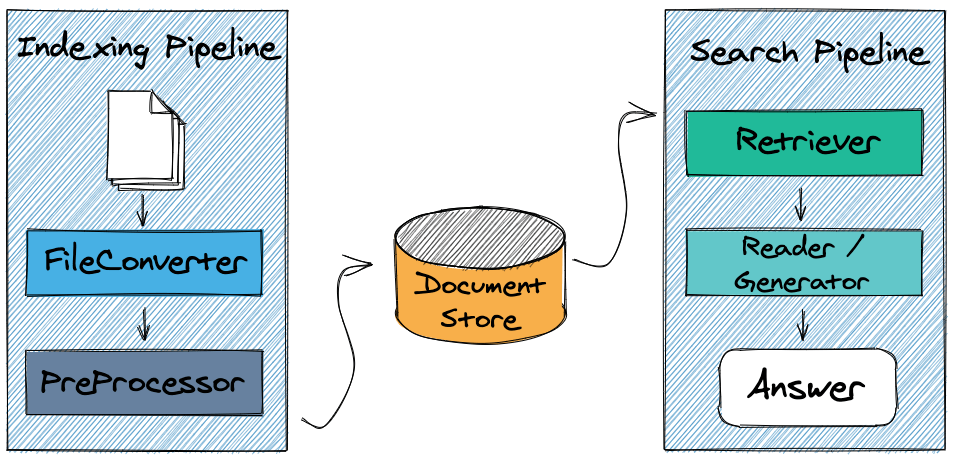
**NLP for Search**: Pick components that perform [retrieval](https://docs.haystack.deepset.ai/docs/retriever), [question answering](https://docs.haystack.deepset.ai/docs/reader), [reranking](https://docs.haystack.deepset.ai/docs/ranker) and much more.

**Latest models**: Utilize all transformer based models (BERT, RoBERTa, MiniLM, DPR) and smoothly switch when new ones get published.

**Flexible databases**: Load data into and query from a range of [databases](https://docs.haystack.deepset.ai/docs/document_store) such as Elasticsearch, Milvus, FAISS, SQL and more.

**Scalability**: [Scale your system](https://docs.haystack.deepset.ai/docs/optimization) to handle millions of documents and deploy them via [REST API](https://docs.haystack.deepset.ai/docs/rest_api).

**Domain adaptation**: All tooling you need to [annotate](https://docs.haystack.deepset.ai/docs/annotation) examples, collect [user-feedback](https://docs.haystack.deepset.ai/docs/domain_adaptation#user-feedback), [evaluate](https://docs.haystack.deepset.ai/docs/evaluation) components and [finetune](https://docs.haystack.deepset.ai/docs/domain_adaptation) models.



url: <https://docs.haystack.deepset.ai/docs>