단어의 의미적 유사도를 응용한 암호 생성기*

최현호[01] 김재웅[1] 이영구[1] 한용구[2]
[1]경희대학교 컴퓨터공학과
[2]메타빌드㈜
abs0lutezer0@khu.ac.kr kju2405@khu.ac.kr yklee@khu.ac.kr ykhan@metabuild.co.kr

Password generator with semantic similarity of words

HyunHo Choi[01] JaeUng Kim[1] Young-Koo Lee[1] YongKoo Han[2]
[1]Department of Computer Science and Engineering, KyungHee University
[2]Metabuild Co.,Ltd.

요 약

문장 간의 유사도를 측정하는 NLP 방식을 활용하여, 단어의 정의문 및 단어 용례의 유사도 검증을 수행한, 유사도가 떨어지는 단어의 조합으로 암호를 생성하여 사용자의 숙지 용이성 및 암호의 안전성을 동시에 확보한다. 암호의 안전성은 기존의 난수화 암호와 동일한 길이의 암호를 생성하여 검증한다. 이를 통해 새로운 형태의 무작위 암호 생성 방법을 제공한다.

1. 서 론

현재, 컴퓨터의 연산 능력이나 각종 처리 기술이 발전함에 따라 강력한 암호화 기술이나 블록체인 기술을 통한 부인 방지 및 변조 방지 기술 등이 꾸준히 개발되고 있다. 물론, 지금은 지문 인식이나 OTP (One Time Password) 등의 다양한 인증 수단이 개발되고 있지만, 아직까지 계정 보안 관리에는 암호 방식이 주로 사용되고 있다. 미국 FBI 등의 연구에서 짧고 복잡한 암호보다 의미적 유사도가 낮은 단어의 긴 조합이 안전하며 암기 측면에서도 더 유용하다고 발표하였다 [1]. 또한 여러 문자와 숫자, 특수문자를 포함한 복잡한 암호보다는 15자 이상의 단어로 이뤄진 긴 문장형으로 된 비밀번호가 보안성이 높다는 분석도 나왔다 [2]. Google Chrome과 같은 사이트 암호 저장을 지원하는 브라우저에서는 난수화한 암호를 생성하는 기능 역시 지원하는데, 정작 해당 기능을 사용하는 Google 계정에 대해서는 해당 보호를 적용하기 어렵다는 단점이 있다. 따라서, 본 논문에서는 단어 유사도 분석 모델을 활용하여 유사도가 낮은 단어들로 암호를 구성하기 위한 방안을 설계한다.

2. 관련 연구

문장이나 문단, 혹은 문서 간의 유사도 분석에 주로 이용되는 유사도 검사는 단어 부문에서는 아직까지 의미적으로 유사도가 매우 높은 단어를 추천하는 시스템이나, 유사도를 기반으로 문장 감성 분석 모델로 활용하여 영화 리뷰 등을 분류하는 등, 높은 유사도 수치를 기반으로 한 서비스가 주로 구상되었으며, 낮은 유사도를 기반으로 한 서비스는 그다지 고안되지 않았다. 따라서, 벡터화 된 유사한 단어들을 한 범주로 묶어주는 유사도 분석 모델을 역이용한다면, 랜덤으로 생성된 단어 간의 유사도 검사가 가능하고, 유사도가 낮은 단어를 이용한다면, 연상하기는 어려우나 난수 암호에 비해 외우기 쉬운 비밀 구절을 생성할 수 있다.

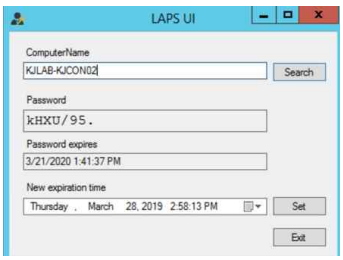
그뿐만 아니라, 앞서 언급한 ETRI의 WiseWordNet의 Open API를 본 연구를 통해 자체 구축할 예정인 모델과 비교하여, 모델의 구현 정도를 검증할 수 있다.

2.1. 단어의 의미적 유사도 분석

NLP(Natural Language Processing, 이하 자연어처리)의 일환으로, 두 문장 속의 단어만을 벡터화하여 유사도를 비교하는 방법이 문장 간 유사도 분석 방법이다. 기존에 구축된 WordNet(프린스턴 영어에 대한 대규모 어휘 데이터베이스)이나 표준국어대사전에 등재된 단어와 그 의미를 토대로 해당 단어에 대한 정의문과 용례를 각각 Sent2Vec 및 Word2Vec 모델로 벡터화하여 단어 의미 유사도를 측정하는 모델에 관한 논문이 2018년 한국콘텐츠학회 학술대회 논문집에 게재되었다 [3].
또한, 위 논문의 연구 배경이 되는 기존 기술인, ETRI가 표준국어대사전을 기반으로 구축한 WiseWordNet은 현재 Open API 형태로 이용할 수 있다.

2.2. 무작위 암호 생성기

현재 계정 보안을 위하여 실제로 운용되고 있는 서비스로는, Windows의 도메인 환경에서 관리자 권한 임시 부여를 위해, 짧은 시간 내에 만료되는 암호를 로컬 시스템의 관리자 계정에 적용하는 LAPS(Local Administrator Password Solution)가 대표적이다. 다만, 랜덤 생성된 암호는 알파벳 대소문자와 특수기호, 숫자를 조합하는데, 유사한 모양의 텍스트로 인해 가독성이 매우 떨어져 생성된 암호를 성공적으로 전달하는 것 역시 쉽지 않다. 다만 암호 만료 기간 및 신규 암호를 주기적으로 갱신되는 그룹 정책에 탑재하여 관리한다는 점이 용이할 뿐이다.



[그림 1] LAPS

*본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학 사업의 연구결과로 수행되었음.
(No. 2017-0-00093)

또한, 앞서 언급한 Google Chrome의 ‘안전한 비밀번호 추천’ 기능이 있다. 계정 가입 절차에 돌입했을 때, 암호 입력 상자에 진입하면, ‘안전한 비밀번호 추천’ 팝업에서 LAPS와 유사한 형태의 난수화된 암호를 제공하고, 해당 암호를 Google 계정에 자동으로 저장하는 방식이다. 해당 계정을 통해 암호를 설정한 시스템의 암호 보안은 상대적으로 안전하겠지만, 암호가 저장된 Google 계정을 사용하지 않으면, 해당 기능을 통해 생성한 암호를 이용하는 사이트는 사실상 이용이 불가능하다는 단점이 있다. 또한, 해당 Google 계정에 대한 암호는 해당 기능을 통해 암호화할 수 없어, 상대적으로 취약한 암호를 사용하거나, 다른 인증 수단을 도입하는 방법밖에 없다.

3. 문제 정의

현재 서비스 중인 암호 생성 서비스는 사용자의 생성 명령 입력 시에 즉각적으로 암호를 출력하는 실시간 서비스이다. 하지만, 암호 생성기로 생성되는 암호는 비밀구절과 달리 난독화, 난수화 작업이 진행되었기 때문에 숙지 난이도가 높고, 가독성이 떨어진다. 따라서, 해당 서비스의 이점을 최대한으로 살리면서, 생성되는 비밀구절의 보안성 및 숙지 용이성을 최대한으로 끌어내도록 구현한다.

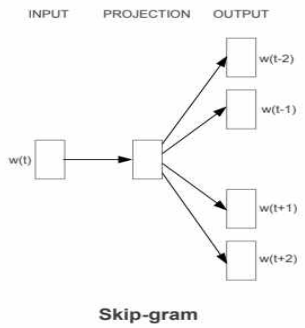
모국어 발음과 실제 입력이 Qwerty 자판(이하 쿼티)을 기반으로 이루어지는 영어나 중국어, 일본어 등과 달리, 한국어는 키보드 입력 방식인 두벌식 표준이 쿼티에 대응되기 때문에, 단어를 쿼티 기반으로 변환했을 때 의미를 한눈에 파악하기 어렵다는 점이 있다. 따라서, 한국어 자판 배열의 비밀구절을 생성하여 암호 생성의 이점으로 활용한다.

생성되는 비밀구절 후보 단어는 일정 유사도 미만의 조건을 만족해야 한다. 또한, 데이터베이스에 등록된 단어는 발음에 따라 숫자 혹은 알파벳으로 치환할 수 있는 글자 (‘이’나 ‘투’라면 2, ‘티’나 ‘비’는 각각 ‘t’와 ‘b’)로 치환한다. 해당 과정을 통해 사전 공격(Dictionary Attack) 취약점을 보완한다. 위 과정을 거친 암호를 시스템의 보안 정책에 부합하는지 검증한다. 검증이 완료된 구절은 숙지를 위해 가공되지 않은 원본과 발음에 따라 치환한 구절을 제공하여 숙지 용이성을 끌어올린다.

4. 단어의 의미적 유사도 기반 암호 생성기 설계

4.1. 단어 유사도 분석 모델

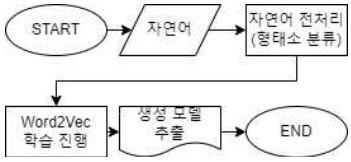
Word2Vec은 단어를 임베딩하여 벡터 유사도 검증을 가능토록 하는 모델이다. 따라서, 우리말샘 사전의 학습데이터로 Word2Vec의 Skip-gram 모델을 학습하여 생성한다. Skip-gram 모델은 하나의 입력값을 기반으로 입력값과 유사한 다수의 출력값을 예측하는 모델이다.



[그림 2] Skip-gram 구조 [4]

학습 데이터는 저작권의 제약을 받지 않으면서, 일상어나 전문용어 등을 망라하는 국립국어원의 우리말샘을 토대로 구축한다. 어휘사전을 통하여 학습을 진행할 경우, 광범위한 범주의 단어 조합으로 인해 적절한 임베딩이 진행되지 않을 수 있는 문제점, 또한 Word2Vec의 동음이의어 임베딩에 있어서의 취약점이 있다는 선행 연구가 있기 때문에 [5], 위키피디아와 같은 백과사전 데이터를 기반으로 한 선행학습 이후 우리말샘 데이터를 추가 학습 시키는 방향, 기존에 선행 학습된 결과가 존재하는 BERT를 통한 추가 학습 등의 요소를 고려한다.

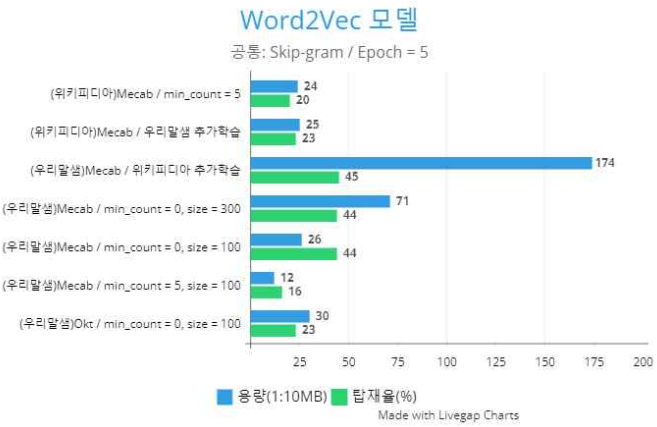
Word2Vec을 이용한 한국어 기반 모델 생성 과정은 [그림 3]과 같다. 자연어 데이터를 반드시 형태소 단위로 분별해야 한다. 데이터의 규모에 맞춰 학습 횟수와 최소 등장 빈도, 단어 벡터 크기 조정 등의 값을 별도로 조절하여 모델의 경량화 혹은 성능 향상을 꾀할 수 있다.



[그림 3] 단어 벡터 생성 과정

본 연구에서는 22년도 3월까지의 위키피디아 게재 문서, 우리말샘 수록 단어 및 그 의미와 용례를 학습에 활용하였다. 형태소 분별을 위한 전처리기로 Mecab 및 Okt를 활용하였다. 단어 탑재율은 임의로 추출한 3개 단어의 유사도를 활용하는 본 연구의 신뢰도에 영향을 끼치며, 따라서 필수적으로 확인해야 하는 항목이다. 따라서, 학습 완료 모델은 암호 생성기에 수록한 524,385개의 단어 DB로 모델 단어 탑재율을 검증하였다.

등장 빈도 5 미만의 단어를 제외하는 설정을 적용한 모델에서의 DB 내 단어 탑재율은 16%에 그쳤다. 그 원인을 옛 우리말이나 방언, 전문용어 등 등장 빈도가 낮다고 예상되는 단어가 학습이 잘되지 않았을 것이라 가정하였다. 따라서 한 번이라도 등장한 단어는 학습이 이루어지도록 조정하고 학습을 진행하였다. 학습 단어 개수 자체의 향상이 이루어졌으나, 탑재율은 약 28%p 개선된 약 44%에 그쳤다. 반복 학습 횟수 증가 및 단어 벡터 크기 증가, 초기 모델 기반 추가 학습 등으로 성능 개선을 시도하였다. 결과는 [그림 4]와 같이 정리하였다.



[그림 4] Word2Vec 모델 용량 및 탑재율

해당 결과를 통해 반복 학습 횟수 증가는 단어 유사도의 정확성에 기여하나, 유의미한 개선은 없다는 것을 알 수 있다. 또한, 단어 벡터 크기는 모델의 용량에 크게 이바지하나, 역시 유의미한 개선은 없음을 알 수 있다. 이후, 위키피디아 모델에는 우리말샘 데이터를, 우리말샘 모델에는 위키피디아 데이터의 추가 학습을 진행하였다. 위키피디아 모델의 경우 min_count 파라미터를 5로 설정하였기 때문에 min_count를 0으로 설정한 우리말샘 모델과의 용량 차이가 크지 않다. 다만, 위키피디아

모델을 기반으로 우리말샘 데이터를 추가 학습을 진행한 결과 순수 위키피디아 모델에 비해 탑재율이 3.3%p 증가하였으나, 여전히 낮은 탑재율을 보였다. 우리말샘 모델에 위키피디아 데이터를 추가 학습시켰을 때, 모델의 크기가 약 1.7GB로, 기존에 비해 약 6.5배 증가하였으나, 탑재율은 단 1%p만이 증가하였다. 그 결과로, 학습 강도나 추가적인 데이터 투입이 본 실험에 유의미한 결론에 다다를 수 없다는 사실을 도출했다. 이후, Okt를 이용하여 우리말샘 데이터를 전처리한 후, 44% 탑재율을 기록한 모델과 동일한 조건에서 학습을 진행하였고, 탑재율이 23% 밖에 미치지 못하였다는 것을 확인하였다. 따라서, 더 이상의 추가 학습은 무의미하다고 판단, 최종적으로 260MB의 용량에 44%의 탑재율을 확보한 모델을 암호 생성기에 도입하였고, 기존 암호 생성기 DB에서 모델에 탑재되지 못한 단어는 제거하여 총 230,500단어를 수록하였다.

모델 학습 과정을 통해 현재의 형태소 분류기는 추가되는 신조어나 전문용어에는 제대로 대응하지 못한다는 결론을 얻을 수 있었고, 추가로 상용 전처리기 중 Mecab의 구동 시간 및 정확성이 높은 이유는 자체 단어사전을 구축하고, 이를 기반으로 형태소를 분류하기 때문임을 알 수 있었다. 따라서, 자주 활용되지 않는 단어들이 Mecab의 단어사전에 수록되지 아니하였고, 그 단어들이 매우 짧은 형태의 문장으로서 인식되어, 형태소 분류기에 감지되는 대로 분해되었다고 추정할 수 있다.

4.2. 단어 기반 암호 강도 향상

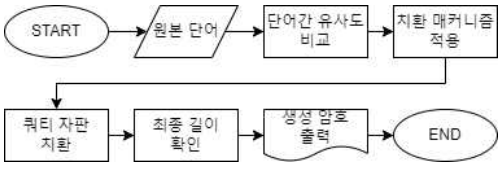
본 연구에선 우리말샘에 등록된 검색어를 기반으로 단어사전을 구축한다. 2022년 3월 기준 등록 단어 750,467개에서 품사 정보가 없는 단어, 옛 한글이 단어 자체 혹은 의미에 포함된 단어, 중복 단어를 제거하여 총 524,385개의 단어를 발음 기반으로 치환하여 DB를 구성한다. 다만, 모델 탑재율 문제로 230,500개의 단어만이 수록되었다.

단어는 쿼티 자판을 기반으로 치환하며, 그 결과의 표본은 [표 1]과 같다.

원본단어	원본->발음	발음->쿼티
에티오펜캅	에t5펜캅	dpt5vpszkq
새끼낳이	새끼낳2	toRlskg2
첨사하다	첨4하다	cja4gkek
베이컨	베2컨	qp2zjs

[표 1] 사전 수록 단어 기반 쿼티 치환 결과

최대 길이 제약이 있는 암호를 요구하기도 하므로, 단어 후보군 추출 시, 추출 단어의 길이를 제한하여 추출할 수 있도록 한다. 또한, 특수기호를 필수적으로 포함해야 하는 경우가 있어, 그 경우 일부 단어의 숫자를 자판에 대응하는 특수기호로 변환하는 선택지를 추가로 제공한다.



[그림 5] 암호 생성 개요

마지막으로, 해당 규칙으로 생성된 암호와 기존의 난수화 암호의 안전성을 비교하기 위해, 모의 환경 구축 후 John the Ripper를 활용한 무작위 대입 공격 및 범용 암호사전 및 자체 구현 암호 목록을 이용하여 사전 기반 공격 시간을 측정한다.

John the Ripper(이하 JTR)는 유닉스 계열 암호 크랙 도구로서, 현재는 Windows, DOS 계열 등 다양한 플랫폼 역시 지원한다. 주로 무차별 대입 공격(Burte Force Attack, 이하 브루트포스 공격)이나 사전 파일 기반 공격(Dictionary Attack, 이하 사전 공격)을 사용하며, 암호 해시값 생성에 사용되는 crypt() 함수를 이용하여 /etc/shadow에 기록된 해시값을 비교하여 암호를 찾아낸다 [6]. 실험 환경 사양은 [표 2]와 같이 정리하였다.

구분	사양
CPU	AMD Ryzen 3400G / 3.7GHz with 4 Cores
RAM	DDR4 3200MHz 16GB * 2
OS	MS Windows 10 / MS WSL2 Kali Linux

[표 2] 실험 및 모델 학습 시스템 사양

해당 환경에서 약 7일간 공격을 진행하였으나, 강화된 암호화 대책으로 인해 브루트포스 공격으로는 유의미한 결과를 도출할 수 없었다. 따라서, 암호 생성기 DB 기반의 암호 사전을 생성하여 크래킹을 진행하였다. 본 실험에 앞서 암호 사전 수록 단일 단어 기반 암호를 설정하여 시험하였고, 명령이 입력되자마자 해독되었다. 하지만, JTR은 단어를 교차하여 조합하는 기능을 지원하지 않아 본 실험에서는 큰 의미가 없었다. 지원한다고 하더라도 최악의 경우 $9.74 * 10^{16}$건의 경우를 비교해야만 한다. 만일 본 암호 생성기가 높은 유사도의 단어 기반이라면, 비교 횟수가 약 $2.73 * 10^{10}$건으로 확연하게 감소한다.

본 생성기를 도입한 시스템이 DES로 암호화되어 1초에 100만 건의 단어 조합을 크래킹에 시도할 수 있다고 할 때, 460,100대의 시스템을 구축하여 최대 59시간 동안 크래킹을 시도해야만 한다. 물론 시스템의 가치에 따라서는 유의미할지 모르나, 일반 사용자 시스템을 위한 비용으로서는 터무니없다. 따라서 시스템 인증을 우회하는 방향이 유리할 것으로 예측된다.

5. 결론 및 향후 연구

암호 복잡도 정책 및 높은 수준의 암호 강도를 만족하면서, 동시에 숙지 및 배포가 용이한 암호를 손쉽게 생성하여 단체 계정 생성 및 배포 시의 초기 암호 설정 및 임시비밀번호 지급 혹은 임시 관리자 권한 부여 시 최소한의 안전성 확보할 수 있을 것으로 기대한다.

향후 연구는 해당 연구의 역으로서 암호를 입력하였을 때 쿼티 기반 암호의 의미적 패턴을 분석하여 암호 강도 측정에 활용하는 방향으로의 진행을 고려할 수 있다. 추가로, 한국어 자연어처리 모델이 신조어나 전문용어에 더욱 잘 대응할 수 있도록 딥러닝 등의 기술을 도입한 유연성 높은 전처리기의 연구가 필요할 것으로 전망한다.

참고문헌

[1] 「Oregon FBI Tech Tuesday: Building a Digital Defense with passwords」, 『FBI』, 2020년 2월 18일.

[2] 「온라인 비밀번호, '이렇게 지어야' 더 효과적」, 『IT조선』, 2020년 2월 26일.

[3] 김호용・이민호・서동민, 「우리말샘 사전을 이용한 단어 의미 유사도 측정 모델 개발」, 『종합학술대회 논문집』 2018년도 춘계, 한국콘텐츠학회, 2018.

[4] Sanket Doshi, 「Skip-Gram: NLP context words prediction algorithm」, 『Towards Data Science』, 2019년 5월 17일.

[5] 이치훈・이연지・이동희, 「사전 학습된 한국어 BERT의 전이학습을 통한 한국어 기계독해 성능개선에 관한 연구」, 『한국IT서비스학회지』 제19권 제5호, 한국IT서비스학회, 2020.

[6] 배유미・정성재・소우영, 「리눅스에 적용된 해시 및 암호화 알고리즘 분석」, 『한국항행학회논문지』 제20권 제1호, 2016.